UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Head Office: Università degli Studi di Padova

Department of Industrial Engineering

Ph.D. COURSE IN INDUSTRIAL ENGINEERING
CURRICULUM: CHEMICAL AND ENVIRONMENTAL ENGINEERING
36th SERIES

INDUSTRY 4.0 IN INDUSTRIAL BIOREFINERIES: IMPROVING PROCESS OPERATIONS BY DATA-DRIVEN AND HYBRID MODELING

Coordinator: Prof. Giulio Rosati
Supervisor: Prof. Massimiliano Barolo

Ph.D. student: Elia Arnese Feffin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial Engineering)
Curriculum: Chemical and Environmental Engineering

at the
University of Padova
2023

Foreword

The research carried out in the PhD project outlined in this Thesis involved the intellectual and financial support of several people and institutions. Most of the research activity has been conducted at the Computer-Aided Process Engineering Laboratory (CAPE-Lab), in the Department of Industrial Engineering of the University of Padova (Padova, IT), under the supervision of Prof. Massimiliano Barolo; relevant contributions to the project were given by Prof. Fabrizio Bezzo and Prof. Pierantonio Facco. The research involved an industrial partner: Novamont S.p.A (Novara, IT); the contribution of Daniele Turati (Novamont S.p.A.) is acknowledged. Part of the research has been conducted during a six-month stay at the Massachusetts Institute of Technology (MIT; Cambridge, MA), under the supervision of Prof. Richard D. Braatz. Financial support for the research discussed in this Thesis has been provided by *Fondazione Cassa di Risparmio di Padova e Rovigo*, *Intesa San Paolo S.p.A.*, *UniSMART – Fondazione Università degli Studi di Padova*, and *Fondazione Ing. Aldo Gini (Padova, IT)*.

All the material reported in this Thesis is original, unless explicit references to studies carried out by other people are indicated. The full list of publications originated from the research described herein is reported below.

CONTRIBUTIONS IN PEER-REVIEWED JOURNALS

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2024). Hybrid modeling of a biorefinery separation process for performance monitoring. *Chemical Engineering Science* **283**, 119413. https://doi.org/10.1016/j.ces.2023.119413.

Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2022). Digital design of new products: accounting for output correlation via a novel algebraic formulation of the latent-variable model inversion problem. *Chemometrics and Intelligent Laboratory Systems* **227**(June), 104610. https://doi.org/10.1016/j.chemolab.2022.104610.

CONTRIBUTIONS IN PEER-REVIEWED JOURNALS (IN PREPARATION)

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023d). *Understanding fouling in an industrial biorefinery membrane separation process by feature-oriented data-driven modeling* [In preparation].

Mohr, F., Arnese-Feffin, E., Barolo, M., and Braatz, R. D. (2023). *Smart process analytics for process monitoring* [In preparation].

CONTRIBUTIONS IN PEER-REVIEWED CONFERENCE PROCEEDINGS

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023c). Troubleshooting high-pressure issues in an industrial biorefinery process by feature-oriented modeling. In: *Computer-Aided Chemical Engineering* **52**, *Proceedings of the 33rd European Symposium on Computer Aided Process Engineering (ESCAPE33)*, 163–168. https://doi.org/10.1016/B978-0-443-15274-0.50027-5.

CONTRIBUTIONS IN CONFERENCE PROCEEDINGS

Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2023a). Estimation of null-space uncertainty in latent-variable model inversion: The case of correlated quality attributes. In: *XI Colloquium Chemometricum Mediterraneum – Book of Abstracts*, 144–145.

CONFERENCE PRESENTATIONS

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023e). Troubleshooting high-pressure issues in an industrial biorefinery process by feature-oriented modeling [Oral presentation]. Presented at: *33rd European Symposium on Computer-Aided Process Engineering – ESCAPE 2023*. June 18–21, Athens (GR).

Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2023b). Estimation of null-space uncertainty in latent-variable model inversion: the case of correlated quality attributes [Poster presentation]. Presented at: *XI Colloquium Chemometricum Mediterraneum – CCM 2023*. June 27–30, Padova (IT).

SOFTWARE AND SIMULATORS

A smart process analytics software for the automated development of fault detection models stems out from the research discussed in this Thesis. The software is open-source and will be made publicly available at the following link upon publication of the relevant journal paper: https://github.com/EliaAF/SmartProcessAnalyticsforProcessMonitoring

Abstract

Biorefineries are innovative processes structured as networks of tightly interconnected plants for sustainable production of energy, fuels, and chemicals using biomass from renewable sources. Individual plants process complex raw materials, characterized by a strong variability in properties, to obtain a limited range of products, often a single one. An integrated use of resources within the network minimizes waste and environmental impact. This is in sharp contrast with traditional refineries that are typically large, centralized plants based on relatively simple, consistent raw materials (oil) and producing a wide range of fuels and chemicals.

Biorefineries based on bioconversion have the traditional structure of a biological process: the operations include biomass preparation, upstream processing, and downstream processing. Despite the key role of biorefineries in circular economy and sustainable production, their effective industrial implementation remains limited. Recent analyses of both the scientific literature and patent databases highlighted that the main challenges in biorefining are both technical and economical. The largest share of the current research is geared towards biomass pre-treatment technologies; research on upstream and downstream processing is ongoing as well, especially regarding process synthesis and design.

Mathematical modeling can significantly aid biorefinery development and operation. Multiple opportunities have been identified for process systems engineering, which are highly relevant to both academia and industry. The great potential of the Industry 4.0 approach, based on advanced data analytics of the large datasets typically produced by modern biorefineries, has been highlighted to improve process operations and performance. This Thesis aspires to make the following contribution: the application of process systems engineering methods and advanced data analytics to support and improve the operations of industrial biorefineries. Specifically, this Thesis focuses on the world's first industrial biorefinery producing 1,4-butanediol by bioconversion of renewable biomass. The plant, located in Bottrighe (IT), is part of the Novamont S.p.A. biorefinery system and represents a remarkable achievement for sustainable, industrial-scale production of building block chemicals.

Two fundamental objectives are pursued in this Thesis.

1. Provide evidence that Industry 4.0 is a precious tool for industrial biorefineries.
2. Contribute to the methodological advancement of data-driven modeling.

The first objective is achieved by developing **digital support systems** to enhance the operations of the industrial biorefinery considered herein. The Industry 4.0 approach is implemented by leveraging advanced data analytics methods to develop **process understanding**, solve **product design** problems, or deploy model-based **process monitoring** systems and **soft sensors**. The second objective is accomplished by pursuing the opportunities offered by the unique industrial

environment in which the project is set and the specific modeling challenges it entails, which could suggest **improvements to existing methods**. A relevant contribution concerns the development of **guidelines to select the most appropriate model** (based on the assumptions the candidate models rely on) for a given task according to the characteristics of the data at hand. The two objectives stated above are achieved in this Thesis observing a general guiding principle: the **incorporation of domain-specific knowledge** in data-driven modeling workflows can significantly enhance the quality, performance, and robustness of the models. Two innovative paradigms are considered in particular: **hybrid modeling** and **feature-oriented modeling**.

Concerning the improvement of operations of the industrial process, a **thorough analysis of the bioconversion step** in the upstream section is presented. The operation relies on an array of fed-batch bioreactors operating in parallel. An Industry 4.0 approach is used to gain process understanding and search for potential differences in performance among the bioreactors. Furthermore, a model of the end-of-batch product quality is developed and leveraged to troubleshoot a decreasing trend of the quality, affecting all the bioreactors. Model interpretation and inversion are used to formulate guidelines to recover the quality of the product.

Regarding the downstream section, a **comprehensive investigation of membrane fouling** taking place in the ultrafiltration unit is discussed. Seven tightly interconnected membrane modules treat the mixture coming from the bioreactors to separate the biomass from the solution containing the biorefinery product. Membranes suffer from fouling issues due to the nature of the feed. However, the current monitoring strategy relies on the visual inspection of profiles of process variables, acquired by operator-read instrumentation installed on the plant, and plagued by high variability due to process settings. Furthermore, only the overall fouling state of the ensemble of modules is monitored, with little to no insight on the state of single membranes.

A **soft sensor for the estimation of resistances** of the seven membranes is proposed. At the heart of the soft sensor is a **hybrid model**: the data-driven element estimates the trans-membrane pressure of each module using process variables as inputs; the trans-membrane pressures serve as inputs to the physics-based element to obtain real-time estimates of membrane resistances. The advantages of monitoring fouling through resistances (rather than process variables) are elucidated, such as resolving effects of reversible and irreversible fouling.

Additionally, a **fouling investigation by feature-oriented modeling** is carried out to identify the process settings most related to this phenomenon. Membrane fouling causes frequent interruptions of operation to clean membranes, leading the process to run in semi-continuous regime. This hinders the application of traditional data analytics methods for process understanding. On the other hand, feature-oriented modeling elegantly solves this problem, while simultaneously enhancing the information on the phenomenon of interest by exploiting process knowledge to design informative features. The results of the analysis confirm the effectiveness of the cleaning policies implemented in the plant and uncover a strong interaction

between reversible and irreversible fouling. This, in turn, offers precious guidelines to improve the maintenance schedule of membranes.

Concerning the advancement of data-driven modeling, a **novel method for algebraic inversion of latent-variable models** is proposed to tackle product design problems. Model inversion can be performed by algebraic manipulation of the model equations, but this requires the quality variables (outputs of the model) to be independent. Therefore, correlated quality variables must be neglected in model calibration. Besides the information loss in modeling, target values cannot be set for the neglected variables, which are not guaranteed to fall within the acceptable quality specifications upon implementation of the inversion solution. Conversely, all quality variables can be considered in the modeling step via the proposed approach, and the numerical issues due to output correlation are addressed only in the inversion step by an optimal regularization with minimal information loss. The advantages of retaining all quality variables are illustrated using two case studies of simulated fermentation processes.

Finally, a framework for the **automatic selection and calibration of data-driven models for fault detection** is proposed. First, rigorously designed criteria are used to assess three key characteristics of the data at hand: nonlinearity of the correlation among variables (equivalent to non-normality of the distribution of data); presence of dynamics in the data; availability of variables describing the product quality. Based on the outcomes of the preliminary data assessment, a subset of appropriate candidate models is selected from the ones in the model library provided alongside the framework. A rigorous model selection and discrimination procedure is then used to calibrate the candidate models, tune their hyperparameters, and select the best performing one for the given dataset. The framework requires data on normal operating conditions only and makes no prior assumption on the nature of faults. The criteria for the preliminary data assessment are validated through rigorous Monte Carlo studies, and the effectiveness of the framework is demonstrated on four case studies: a simulated linear, static dataset; the Tennessee Eastman Process simulator; a simulation of a process for continuous filtering and drying of paracetamol; data from an industrial metal etching process for semiconductor manufacturing. The proposed framework successfully identifies the most appropriate model (among those included in the library) in all case studies, based on the fault detection performance on data from faulty conditions (not used for model calibration).

The studies presented in this Thesis provide strong evidence of the value of the Industry 4.0 approach and represent significant steps in the digitalization of industrial biorefineries. Advanced data analytics methods can disclose valuable information on the complex operations implemented in modern biorefineries, such as bioreactor arrays and membrane filtration units. Digital support systems can significantly enhance operations. Novel frameworks for model selection and evaluation can make such sophisticated methods readily available to practitioners. Overall, the results presented in this Thesis are expected to promote the adoption of the Industry 4.0 approach in challenging industrial environments, such as biorefineries.

Table of contents

List of symbols

ACRONYMS

ACE	=	Alternating conditional expectation
ACF	=	Autocorrelation function
ANOVA	=	Analysis of variances
AutoML	=	Automated machine learning
BDO	=	1,4-butanediol
BWU	=	Batch-wise unfolding
CCA	=	Canonical correlation analysis
CCF	=	Cross-correlation function
CER	=	Carbon dioxide evolution rate
CI	=	Confidence interval
ContCarSim	=	Continuous carousel simulator
CV	=	Canonical variable
CVA	=	Canonical variate analysis
DKPCA	=	Dynamic kernel principal component analysis
DKPLS	=	Dynamic kernel partial least-squares
DI	=	Direct inversion
DPCA	=	Dynamic principal component analysis
DPLS	=	Dynamic partial least-squares
DO	=	Dissolved oxygen
DOF	=	Degrees of freedom
EU	=	European Union
EV	=	Explained variance
IEA	=	International Energy Agency
IQR	=	Inter-quartile range
JYPLS	=	Joint-Y partial least-squares
KDE	=	Kernel density estimation
KPCA	=	Kernel principal component analysis
KPLS	=	Kernel partial least-squares
LV	=	Latent-variable
LVMI	=	Latent-variable model inversion
MSE	=	Mean-squared error
NNPCA	=	Neural network principal component analysis

NOC	=	Normal operating conditions
OCC	=	One-class classification
OD	=	Optical density
OUR	=	Oxygen uptake rate
PACF	=	Partial autocorrelation function
PC	=	Principal component
PCA	=	Principal component analysis
PCR	=	Principal component regression
PDF	=	Probability density function
PID	=	Proportional-integral-derivative
PLS	=	Partial least-squares
PLSDA	=	Partial least-squares discriminant analysis
PSE	=	Process systems engineering
RBF	=	Radial basis function
RDI	=	Regularized direct inversion
SEC	=	Standard error in calibration
SPA	=	Smart process analytics
SPAfPM	=	Smart process analytics for process monitoring
SPC	=	Statistical process control
SVD	=	Singular values decomposition
SVDD	=	Support vector data description
TEP	=	Tennessee Eastman Process
TMP	=	Trans-membrane pressure
USD	=	United States dollar
VCR	=	Volume conversion ratio
VWU	=	Variable-wise unfolding

Chapter 1

Motivation and literature review

This chapter introduces the scope of the Thesis. Concepts on biorefineries and the sustainable production of 1,4-butanediol (BDO), the chemical of interest in this context, are discussed first. The potential of process systems engineering, in particular data-driven modeling in the form of the Industry 4.0 approach, in this field is highlighted in a literature review. Finally, the research objectives of this Thesis are stated, and a roadmap of this document is outlined.

1.1 The biorefinery concept

1.1.1 Biorefineries: concept and introduction

Biorefineries are facilities integrating sustainable biomass conversion processes and equipment to output a range of products (Cherubini, 2010; Taylor, 2008; Velidandi et al., 2023), among which fuels, such as ethanol (Delgenes, 1996) or biodiesel (McCurdy et al., 2014), and building-block chemicals, such as succinic acid (Mancini et al., 2020) or BDO (Burgard et al., 2016). Biorefinery processes are designed not only according to profitability purposes, but also considering sustainability, social impact, and circular economy concepts (Ioannidou et al., 2020). This is usually achieved integrating traditional process synthesis and design methods with proper sustainability metrics (Sikdar, 2003) and life cycle assessment, including its environmental and social variations (Julio et al., 2017).

Substances traditionally derived from petroleum can be produced in a sustainable way in biorefineries (Martín et al., 2013), which is due to the main feedstock[1]: biomass. While both oil and biomass represent carbon-rich raw materials, there are significant differences between traditional refineries and biorefineries (Attard et al., 2020; Cherubini, 2010). Traditional refineries are usually large, centralized plants processing a consistent, simple raw material available year-round from specific locations (therefore needing massive transportation systems) and aimed at the production of a wide range of fuels and chemicals. On the other hand, industrial biorefineries are (expected to) develop as a decentralized, interconnected, and integrated network of plants producing a small range of products (within single plants) by processing local, renewable raw materials subject to seasonal variability (for example agricultural crops);

[1] In this Thesis, the term "feedstock" is intended as defined by Cherubini (2010), that is 'the raw material used in the biorefinery'.

further variability is added by the variety of feedstock available for biorefining (Cherubini, 2010). The most relevant difference is, however, the paradigm-shift in how resources are used. In the words of Cherubini (2010): «Biorefinery represents a change from the traditional oil refinery based on large exploitation of natural resources and large waste production towards integrated systems in which all resources are used». A brief summary of the main differences between traditional refineries and biorefineries is reported in Table 1.1.

Table 1.1. *Comparison between traditional refineries and biorefineries.*

Characteristics	Traditional refinery	Biorefinery
Plant size	Large plant	Small plant
Plant structure	Centralized location	Distributed network
Raw materials	Fossil oil	Renewable biomass
Raw materials properties	Relatively simple and consistent	Complex and strongly variable
Raw material origin	From centralized locations	Locally sourced
Raw material availability	Year-round	Subject to seasonality
Process operations	Separations and simple chemical conversions	Complex pre-treatments, conversions, and separations
Energy demand	Energy-intensive process	Energy-efficient process
Use of resources	Linear with large waste fraction	Integrated with re-use and valorization of waste

Biorefineries can be distinguished according to their feedstock. An established classification defines three generations of biorefineries (Martín et al., 2013):

- the first generation uses food-related biomass, as sugar, corn, or wheat (Martin, 2010);
- the second generation is based on biomass discarded as waste from other processes, as wood waste or exhaust frying oil (Prunescu, 2015);
- the third generation relies on algae (Alam et al., 2015).

Fourth generation biorefineries are also being discussed in the literature. However, the designation of the feedstock is still unclear. A significant research is being devoted to fungal biomass as a potential fourth generation feedstock (Amerit et al., 2023; Varriale et al., 2023). However, other biomass categories have been proposed to the same end (even though some are already included in three generations mentioned above): vegetable oils (Gavrilescu, 2014), green crops such as grass (Gaffey et al., 2023), lignocellulose (Yadav et al., 2023), residues of coffee (Strieder et al., 2023) and tea (Kumar et al., 2023), or food waste (Ioannidou et al., 2020). The selection of the feedstock is critical, as it determines most of the main pre-treatment technologies for extraction of/conversion to relevant compounds. Significant research on this field is ongoing (Attard et al., 2020); however, four main so-called biorefinery platforms have been proposed and are widely accepted (Cherubini, 2010; Gavrilescu, 2014; Ubando et al.,

2020): thermal conversion, mechanical conversion, chemical conversion, and biological conversion. While each one of these technologies can process different biomasses and yield different products, as depicted in Figure 1.1, the focus of this Thesis is on biorefineries operating biological conversion of biomass into biochemicals, specifically by fermentation of sugars derived from enzymatic hydrolysis of biomass. Readers interested on different platforms and products are referred to the studies cited above.

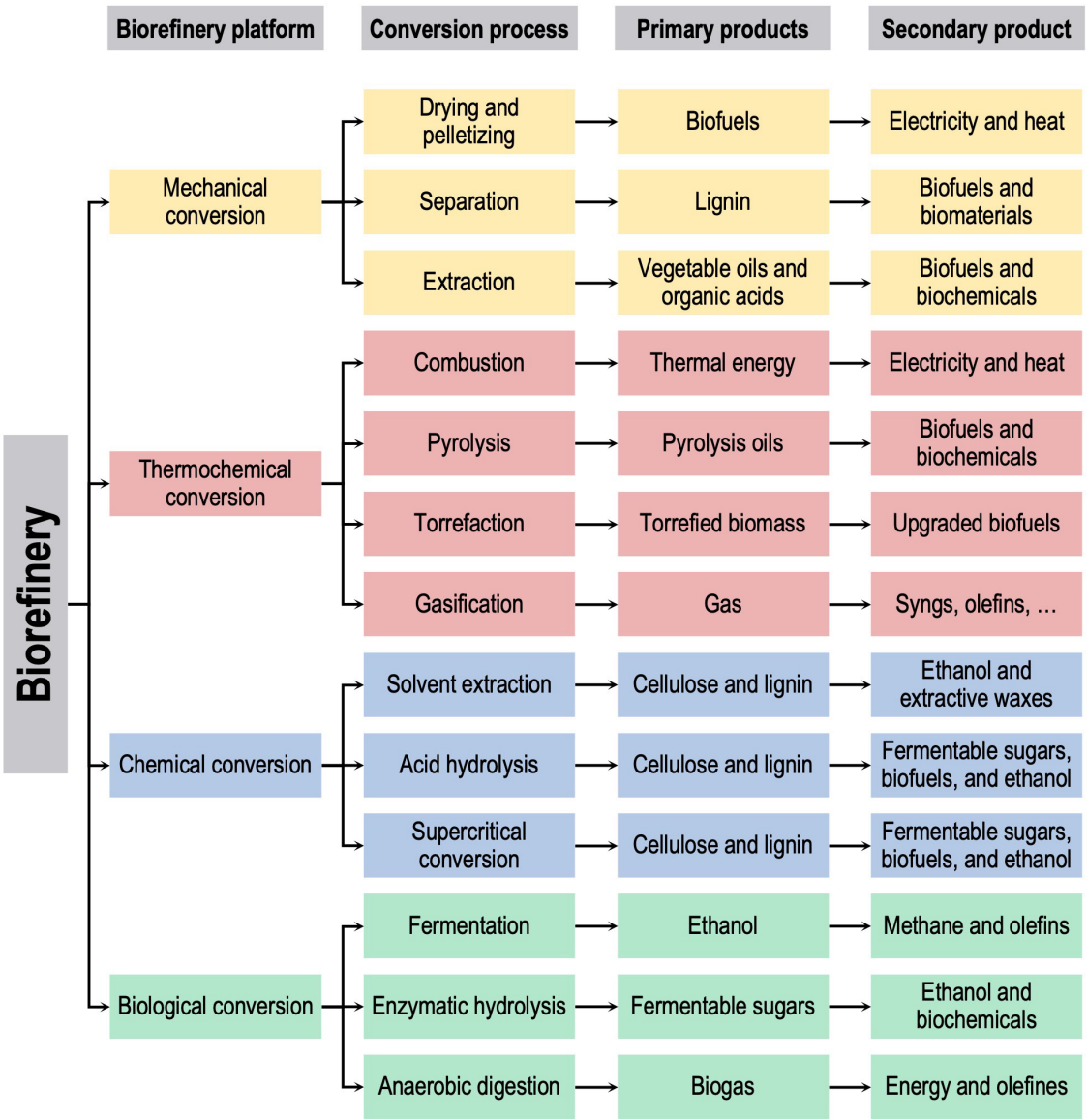


Figure 1.1. *Schematic representation of biorefinery platforms, biomass conversion processes, primary products, and potential secondary products. Adapted from Ubando et al. (2020) and Gavrilescu (2014).*

1.1.2 Process structure

Biological conversion has gained a wide attention recently (Woodley, 2020) due to the potential to produce a wide range of chemicals, besides the widely established fermentative ethanol (Cherubini, 2010; Cuellar et al., 2020; Rosales-Calderon et al., 2019). Advancements in genetic

engineering are the main enablers to this end (Barton et al., 2015; Burgard et al., 2016; Cheng et al., 2021; Choi et al., 2016; Lee et al., 2019; Noorman et al., 2017; Wehrs et al., 2019; Woodley, 2020; Yim et al., 2011). This endows biorefineries with the traditional structure of bio-based processes; operations typically include: feedstock and media preparation, preliminary growth of microorganisms, and large-scale conversion in bioreactors in the upstream section; broth sterilization, cell separation, and product recovery and purification in the downstream section (Bähner et al., 2021).

With reference to biorefineries operating by fermentation, the first step of the process is biomass preparation to obtain suitable substrates for the microorganisms, most commonly fermentable sugars. Biomass categories that are naturally rich in sugars, such as sugar cane, can be processed directly in bioreactors (Ubando et al., 2020). On the other hand, starchy biomasses, such as corn, need to be treated to break down the biopolymer chains to their fundamental constituents; enzymatic hydrolysis is the preferred path for such operation (Delbecq et al., 2018). The culture medium has to be prepared as well, and its optimization is of paramount importance to ensure that expensive raw materials are not wasted and to avoid increasing the separation burden on the downstream section (Burgard et al., 2016).

After biomass preparation, sugars are fed to bioreactors. Fermentation is the most common upstream operation in bioconversion-based biorefineries (Woodley, 2020), typically operated in (fed-)batch mode (Bähner et al., 2021). With focus on the production of chemicals, a range of relevant building blocks can be obtained using natural microorganisms (Cherubini, 2010). However, genetic engineering has given a tremendous impulse to fermentation-based bioconversion by solving problems such as low yield or productivity (Woodley, 2020), and even enabling the production of "unnatural" chemicals, for example succinic acid (Choi et al., 2016) and BDO (Burgard et al., 2016; Yim et al., 2011); a large number of other molecules could be potentially produced by fermentation thanks to the recent developments in genetic and metabolic engineering (Lee et al., 2019). On the other hand, traditional, single-step fermentation is not the only available technology for biological conversion: the potential of alternative processes has been recently recognized, most notably microbial and enzymatic bio-catalysis (Woodley, 2020), and two-stage fermentation (Burg et al., 2016).

The outlet of the fermentation reactor is typically a complex aqueous solution containing cells and a mixture of substances, among which the desired product. After sterilization, the mixture enters the downstream process for product recovery and purification. Due to the typically low concentration of the main product in fermentation broths, processing costs in this section are usually high (Cuellar et al., 2020; Martín et al., 2013), ranging between 40% and 60% of the total processing cost, with peaks of 80% in some special cases; energy and utilities are the main cost components (Bähner et al., 2021; Cuellar et al., 2020).

While fermentation is a complex process taking place in a relatively simple unit, product purification is achieved by a complex sequence of units implementing relatively simple

operations (compared to fermentation), in fact a broad range of technologies is available (Hatti-Kaul, 2010). However, the operations to be performed can be classified in four main steps, exemplified here with focus on BDO (Burgard et al., 2016; Cuellar et al., 2020):

- cell removal, usually achieved by filtration on porous membranes (microfiltration and ultrafiltration);
- preliminary product purification by removal of other compounds and salts, achieved by dense membrane separations (nanofiltration) and ion-exchange chromatography;
- product concentration by dewatering in evaporators;
- final product purification, achieved by traditional distillation.

Membrane separation processes have been highlighted as a particularly promising technology for cell removal after fermentation (Bähner et al., 2021; Cuellar et al., 2020; Prochaska et al., 2018). In the context of biorefineries, membrane technologies have been widely studied and are becoming increasingly relevant (Abels et al., 2013; Carstensen et al., 2012; Ennaceri et al., 2022; Gerardo et al., 2014; Saha et al., 2017) thanks to their better scalability and lower operating costs compared to thermal separation processes (Ennaceri et al., 2022; Gerardo et al., 2014; Jiang et al., 2013; Saha et al., 2017). Pressure-driven membrane separation processes, for example ultrafiltration, are particularly popular to separate biomass from the fermentation products when membrane-based operations are employed to this end (Rudolph et al., 2019).
While, in principle, all the aforementioned downstream operations can operate continuously, they are most commonly run in batch or semi-continuous modes. This is due to the (fed-)batch operating regime of upstream fermentation, the product of which is available only after batch discharge, (Bähner et al., 2021), and to phenomena such as resin depletion (Hatti-Kaul, 2010; Zydney, 2016) and membrane fouling (Huang et al., 2021; Mancini et al., 2020; Prochaska et al., 2018; Shi et al., 2014). Both problems could be solved or mitigated by employing arrays of parallel units in cyclic operation (Zydney, 2016). This solution is already widely adopted in upstream processing of bioprocesses to make the outlet of bioreactors available continuously, even at the cost of a complicated scheduling (Bähner et al., 2021), while it is less common in the downstream processing (Zydney, 2016), where operation-cleaning/regeneration cycles are used preferentially (Hatti-Kaul, 2010; Shi et al., 2014).

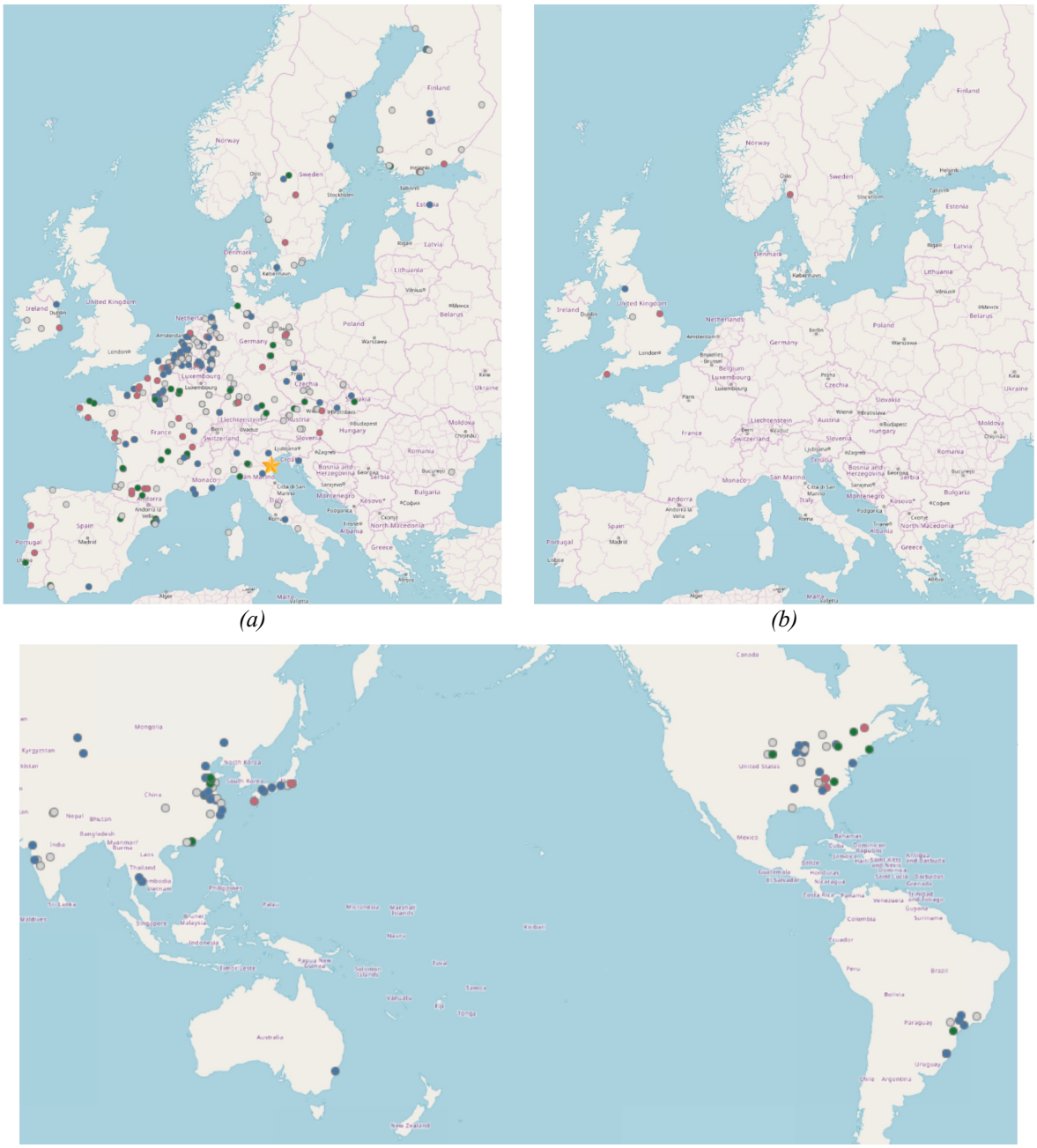
1.1.3 Industrial implementation of biorefineries

A wide spectrum of fuels and commodity chemicals can be produced by fermentation of sugars and biomass in single-product biorefineries (Cherubini, 2010) or in integrated, multiproduct plants (Rosales-Calderon et al., 2019). Many technologies and microorganisms have already reached the maturity for industrial-scale production (Cuellar et al., 2020), and the potential of genetic engineering is expected to enlarge the pool of biorefinery products (Lee et al., 2019). Furthermore, besides the obvious monetary purposes, the transition from the linear economy model to the circular economy model is of paramount importance for efficient exploitation and

re-use of resources, and for contrasting the already dramatic effects of climate change (Attard et al., 2020; Ioannidou et al., 2020; Ubando et al., 2020). Last but not least, policy makers, such as the European Union (EU), committed to regulate, oversee, and partially fund the industrial and academic development of novel, environment-friendly production means like biorefineries (European Commission, Directorate General for Research and Innovation et al., 2021; European Commission, Joint Research Centre et al., 2021, 2018); furthermore, international agencies, as the International Energy Agency (IEA), dedicated to track and ease the realization of operating plants (IEA Bioenergy: Task 42 Biorefining in a circular economy et al., 2022).
All these factors gave a strong impulse to research, development, and implementation of novel technologies for production of chemicals traditionally derived from oil. Ultimately, this resulted in the construction and startup of many plants qualified as biorefineries: according to EU data updated to December 10, 2022 (European Commission, Joint Research Centre et al., 2021), 408 biorefineries are already operating worldwide: 298 within the EU (Baldoni et al., 2021a) and 110 in extra-EU Countries (Baldoni et al., 2021b); in 2017, 224 biorefineries were operating within the EU (Nova Institute, 2017), which testifies the growing industrial interest in this field. However, note that EU data include all plants qualified as biorefineries, with varying degrees of production capacity and technology maturity (European Commission, Joint Research Centre et al., 2021). Filtering the data to retain only biorefinery employing commercial technologies (pathways A, B, and C) producing chemical and liquid biofuels yields 122 biorefineries within the EU (Baldoni et al., 2021a) and 79 outside the EU (Baldoni et al., 2021b), which still include also small-scale plants (pilot and demonstration); therefore, the number of industrial-scale plants is expected to be lower than the one reported above. A general overview of the geographic distribution of biorefineries is reported in Figure 1.2. The reader is referred to EU data (Baldoni et al., 2021a, 2021b; European Commission, Joint Research Centre et al., 2021) for additional details.

1.1.4 The world's first industrial biorefinery for 1,4-butanediol production

Among the currently operating plants, a significant one is represented by the world's first industrial biorefinery for the production of BDO by bioconversion of renewable raw materials, located in Bottrighe (IT), and operated by Mater-Biotech S.p.A. and Novamont S.p.A. (Novamont S.p.A., 2016).
BDO is a fundamental building block chemical with a remarkable market. According to a report by Grand View Research (2022), the estimated value of the BDO market was 7.87 billion United States dollars (USD) in 2022 and is forecasted to almost double in 2030, reaching 14.66 billion USD. The Asia-Pacific region leads the BDO manufacturing market. BDO is used to produce a variety of chemicals, tetrahydrofuran being its main derivative, and polymers, among which polybutylene terephthalate and polyurethanes. Solvents, such as the γ-btyrolactone, can be produce from BDO as well (Rosales-Calderon et al., 2019).



(a) *(b)*

(c)

Figure 1.2. *Overview of the geographic distribution of biorefineries (a) in Countries within the EU (Baldoni et al., 2021a), (b) in Countries outside of the EU but still in Europe (Baldoni et al., 2021b), and (c) in the rest of the world (Baldoni et al., 2021b). Dots are colored according to the product category of a plant: blue for chemicals, red for composites and fibers, green for other products, and grey for multiple products. The biorefinery considered in this Thesis and described in Section 1.1.4 is marked with a golden star in (a).*

BDO is traditionally derived from fossil raw materials with energy-intensive processes having a significant impact on the environment (Rosales-Calderon et al., 2019). The two-stage Reppe process is the most widespread route for commercial production: formaldehyde and acetylene react to form 1,4-butynediol in the first stage; hydrogenation completes the synthesis in the second stage. Alternative, yet still oil-based, processes exist. In the Mitsubishi Chemicals

process, butadiene is treated with acetic acid by oxidative acetoxylation, then hydrogenated and hydrolyzed to BDO. The Arco Chemicals route starts from propylene oxide, which is first isomerized to allyl alcohol, then hydroformylated with syngas (a mixture of hydrogen and carbon monoxide) to 4-hydroxybutyraldehyde, and finally hydrogenated to BDO. Processes based on maleic anhydride exist as well, such as the Davy Technology diesterification-hydrogenation, or the British Petroleum direct esterification.

More recently, sustainable, energy-efficient processes have been developed to produce BDO from renewable resources. A catalytic route to produce BDO by direct hydrogenation of succinic acid, a compound that can be obtained by fermentation (Cherubini, 2010) at industrial scale (Cuellar et al., 2020; López-Garzón et al., 2014), has been proposed (Baidya et al., 2019; Kang et al., 2015). A significant achievement is represented by the Genomatica process, where BDO is produced in a single-step bioconversion of sugars operated by genetically engineered microorganisms (Burgard et al., 2016; Yim et al., 2011).

The latter technology has been acknowledged as a huge achievement for genetic engineering and bio-based production systems (Wehrs et al., 2019), besides being the only commercial process to produce BDO directly from sugars (Bodor et al., 2019); it also prompted significant research in the field of biorefinery process design (Noorman et al., 2017; Teh et al., 2019). Techno-economic analyses of the process proved its economic feasibility and competitiveness with routes based on fossil raw materials, highlighting that the major capital costs are due to the bioconversion step, while membrane separation processes in the downstream represent the major operating cost (Satam et al., 2019). Concerning the environmental impact, the equivalent emission of CO_2 (as kilograms of CO_2 emitted per kilograms of product formed) was estimated to be between 52% and 82% lower with respect to the traditional Reppe process, while the usage of non-renewable energy from fossil fuels was estimated to be reduced between 67% and 72% (Burgard et al., 2016; Forte et al., 2016). Equivalent processes, to be implemented in second generation biorefineries, are under study as well (De Bari et al., 2020).

The Genomatica technology is at the heart of the aforementioned Novamont biorefinery, the first one implementing this process at commercial scale (Novamont S.p.A., 2016; Silva et al., 2020). The plant, which falls within the definition of first generation biorefinery as it processes entirely renewable sugars by bioconversion, has been designed with a production capacity of 30000 metric tons of BDO per year (IEA Bioenergy: Task 42 Biorefining in a circular economy et al., 2022; Novamont S.p.A., 2016). In the upstream section, the bioconversion is conducted in “micro-aerobic” conditions (Burgard et al., 2016), which entails the need for a precise control of the dissolved oxygen (DO) level. Downstream of the bioreactor, the mixture is first sterilized, then cells, high molecular weight compounds, and salts are removed by means of membrane filtration and ion-exchange chromatography. These operations belong to the so-called “purification” step, that is followed by the “refining” step, where dewatering and distillation complete the downstream train (Burgard et al., 2016; Cuellar et al., 2020).

Furthermore, the plant is equipped with a massive data acquisition and storage systems, providing plenty of real-time measurements to monitor and control the process. A qualitative scheme of the process is reported in Figure 1.3.

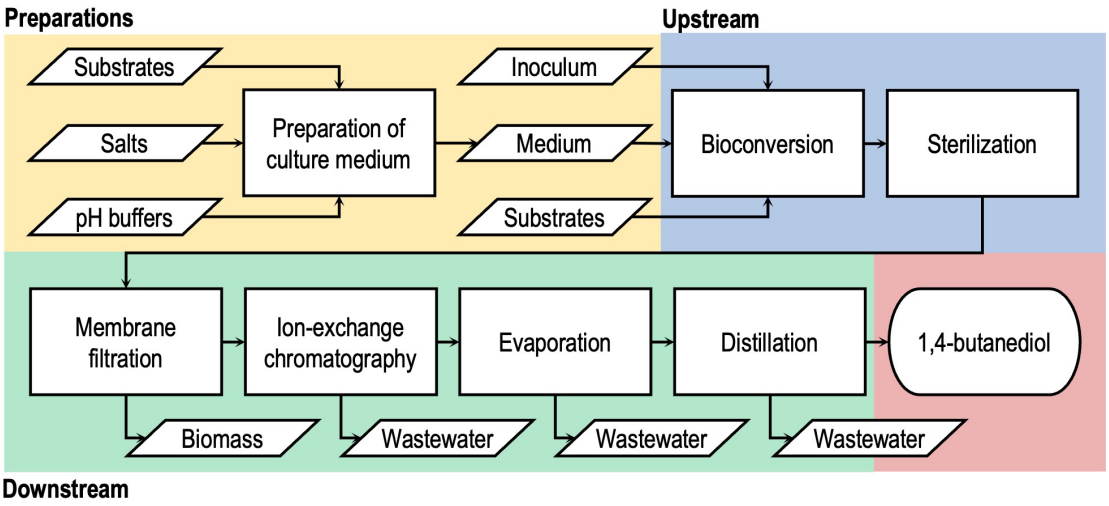


Figure 1.3. *Simplified block flow diagram of the industrial biorefinery process for the production of BDO from renewable biomass, operated by Mater Biotech S.p.A. in Italy.*

The industrial BDO biorefinery described above is the object of this Thesis, carried out as a collaboration between Novamont S.p.A. and the Computer Aided Process Engineering Laboratory (CAPE-Lab) of the University of Padova (Italy).

1.1.5 Research trends in biorefining

In the context of circular economy and production of fuels and chemicals from sustainable resources with an eye for environmental and social impacts (Attard et al., 2020; Ioannidou et al., 2020; Ubando et al., 2020), the potential of biorefineries has been recognized in the last decades (Cherubini, 2010; Taylor, 2008). Such potential developed and materialized in many processes achieving commercial maturity (Cuellar et al., 2020; Rosales-Calderon et al., 2019) and in a remarkable number of biorefineries being built and operated around the world (Baldoni et al., 2021a, 2021b). Despite these tremendous achievements in the fields of circular economy, sustainable production, and biorefining, significant research is still ongoing and future outlooks are being given by academia (Barragán-Ocaña et al., 2023), industry (Bähner et al., 2021), and policy makers (European Commission, Directorate General for Research and Innovation et al., 2021). An overview of some relevant research topics is reported in Figure 1.4.

Recent analyses of both the scientific literature and patent databases (Barragán-Ocaña et al., 2023) highlighted that the main challenges associated with biorefineries are both technical and economical. Holistic and multidisciplinary approaches are fundamental to tackle such issues. As the raw materials for fermentation-based biorefineries are sugars, efforts are being devoted to the development of technologies to pre-treat biomass derived from non-food sources.

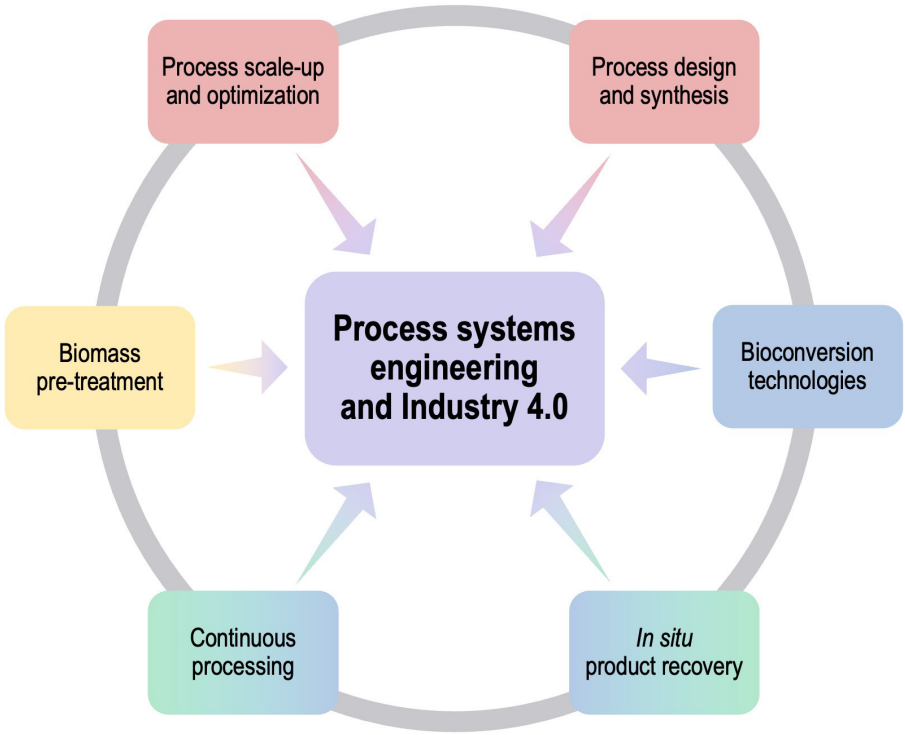


Figure 1.4. *Significant research topics concerning biorefineries.*

Variability in chemical composition, both geographical and seasonal, entails high process complexity and requires special unit operations, which are object of significant research (Attard et al., 2020; Barragán-Ocaña et al., 2023). This also calls for novel and systematic approaches for process synthesis and design (Ubando et al., 2020): work ongoing in this field regards new process design workflows (Noorman et al., 2017), systematic evaluation of potential technologies (Martín et al., 2013), and incorporation of environmental and social objectives in the traditional, profit-oriented process design paradigm (Julio et al., 2017; Teh et al., 2019).
Concerning the upstream process, where classic single-step (fed-)batch fermentation is still the dominant production regime (Bähner et al., 2021), research is focusing on alternative bioconversion technologies, for example microbial bio-catalysis, enzymatic bio-catalysis (Woodley, 2020), and two-stage fermentation (Burg et al., 2016). Bioengineering tools are powerful assets to this end, but also to broaden the range of products that can be manufactured in biorefineries (Lee et al., 2019). Scale-up, a well-known issue in biological processes, is also challenging (Woodley, 2020). Following the trend of the pharmaceutical industry, continuous processing represents an attractive solution for productivity and efficiency (Cuellar et al., 2018), even though the applicability to biorefineries is still limited by genetic stability of engineered strains (Noorman et al., 2017) and contaminations (Bähner et al., 2021).
Challenges exists for the downstream as well (Cuellar et al., 2020). Research is focusing on numerous topics, among which: high throughput experimentation platforms; understanding of effects of the composition of fermentation media on downstream operations; cell disruption for recovery of intracellular compounds; membrane separation processes; dividing wall columns

for process intensification and to solve complex separation problems. Improvement of operating modes of the downstream, currently limited to batch or semi-continuous modes by upstream operation and phenomena such as membrane fouling and resin capacity depletion, is also receiving a wide attention (Zydney, 2016). Finally, a significant share of research is aimed at "jointing" the upstream and downstream processes by *in situ* product recovery, for example by membrane bioreactors (Carstensen et al., 2012; Cuellar et al., 2018; Rudolph et al., 2019).
A common denominator can be extracted by all the studies mentioned above: mathematical modeling has a remarkable potential to aid biorefinery development and operation. Multiple opportunities have been identified in the field process systems engineering (PSE), which are highly relevant to both academia and industry (Bähner et al., 2021). In particular, advanced data analytics, based on massive data historians produced by modern biorefineries, has been highlighted as a tool with a great potential to improve operation and performance of biorefinery plants and processes, and to enhance scale-up and scale-down in process development (Cuellar et al., 2020; Culaba et al., 2022; Velidandi et al., 2023). The contribution of this Thesis falls within this scope: the application of PSE methods and advanced data analytics to develop knowledge on and support the operation of the world's first industrial biorefinery for the production of BDO. A detailed review of the potential of PSE and data analytics is presented in the next Section.

1.2 Mathematical modeling to support biorefinery operation

Mathematical modeling, especially data-driven approaches such as statistical, machine, and deep learning, have been acknowledged as essential tools to support the development of smart biorefineries and hasten the deployment of cutting-edge technologies (Bagheri et al., 2019; Cuellar et al., 2020; Culaba et al., 2022; Helleckes et al., 2022; Velidandi et al., 2023). Novel biorefineries have in fact a powerful ally to maximize the benefits that could potentially originate from such opportunities: the Industry 4.0 paradigm.

1.2.1 Industry 4.0 relevance to industrial biorefineries

The concept of Industry 4.0 refers to the fourth industrial revolution, that is taking place in our time. This is being driven by three key enablers (Reis et al., 2017, 2021b): availability of massive datasets (the so-called big data); maturity of computational infrastructures; power of modern data analytics methods. Similarly to the previous three revolutions (the first one driven by coal and steam, the second one by electricity, and the third one by the advent of computers), the industry is experiencing a significant expansion and is making its way toward the full implementation of the so-called smart manufacturing systems (Reis et al., 2018). Biorefineries are already taking advantage of these new concepts, as testified by the multitude of successful applications of data-driven modeling described in numerous literature reviews (Bagheri et al.,

2019; Culaba et al., 2022; Helleckes et al., 2022; Pomeroy et al., 2022; Velidandi et al., 2023). However, the full potential of such methodologies is far from being achieved. In particular, two relevant points are worth mentioning regarding the current state of data-driven modeling in biorefineries:

- the application of data-driven modeling to real, industrial plants and data is limited;
- highly complex, heavily nonlinear, non-interpretable methods are used by default in the literature, with little to no attempt to explore simpler, more robust approaches first.

While the first point is justified by the still limited presence of real industrial biorefinery plants, the second point may be troublesome for multiple reasons. Logically, one should first explore simpler modeling methodologies according to some complexity scale (Rendall et al., 2019), as complex models featuring more degrees of freedom have higher risk of overfitting, yielding misleading conclusions and poor generalization performance (Reis et al., 2018; Rendall et al., 2019; Sun et al., 2021). Furthermore, industrial data often lack two fundamental prerequisites for the proper calibration of complex models: quantity and quality (Cuellar et al., 2020; Reis et al., 2018; Sun et al., 2021). Unawareness of these factors can seriously limit the benefits of an Industry 4.0 approach, especially in industrial scenarios, where it matters the most. Approaches to measure the quality of information that can be obtained from a data-driven study with given objectives and allocated resources can make a difference in such cases (Reis et al., 2018).

This Thesis aspires to make up for the aforementioned limitations. Examples of tasks that can be accomplished by adopting an Industry 4.0 approach to support the operation of industrial biorefineries are discussed in the rest of this Section and are schematically represented in Figure 1.5. A review of some fundamental contributions of data-driven modeling to biorefinery and biorefinery-like scenarios, such as batch reactors, bio-based processes, and membrane separation processes, is presented. Particular attention is devoted to simple approaches that proved to be effective in industrial case studies; an overview of more complex machine learning techniques can be found in recent literature reviews (Bagheri et al., 2019; Culaba et al., 2022; Helleckes et al., 2022; Pomeroy et al., 2022; Velidandi et al., 2023). Note that the purpose of this Section is just to gather relevant contributions: mathematical descriptions of some of the approaches that will be mentioned throughout this Section is given in Chapter 2, while research objectives of this Thesis are properly stated in Section 1.3. The studies through which the research objectives are achieved are outlined in Section 1.4.

1.2.2 Development of knowledge by exploratory data analytics

One ubiquitous feature of industrial process data is correlation among variables (Kourti et al., 1995; Wise et al., 1996). This simple feature can cause many numerical issues, preventing the use of classic regression methods, as ordinary least-squares regression (based on the assumption of independent variables), and causing identifiability issues in more complex models. On the other hand, there exist approaches able to cope by design with correlation among variables,

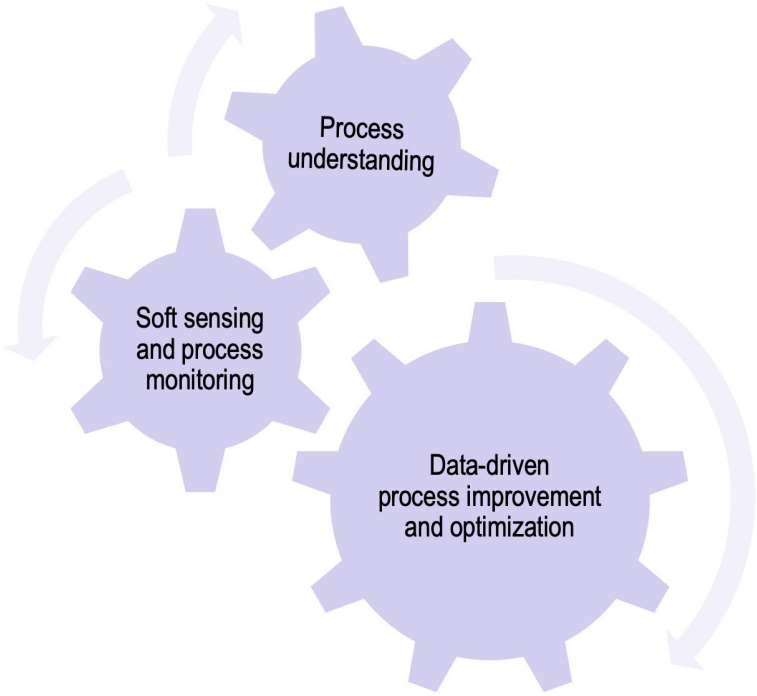


Figure 1.5. *Potential benefits of the application of the Industry 4.0 approach to industrial biorefineries.*

such as variables selection combined with traditional methods, penalized regression, latent-variable models, and tree-based ensemble models (Reis et al., 2018; Rendall et al., 2017b). Latent-variable methods (Burnham et al., 1999), such as principal component analysis (PCA) and partial least-squares (PLS) regression, are particularly attractive due to their architectural simplicity, ability to deal with massive numbers of variables, computational efficiency, ease of implementation, and straightforward interpretability. Regarding the latter aspect, the fundamental assumption of latent-variable models is that the process is dominated by a limited number of driving forces (latent variables) to which measurable variables are just proxies (Kourti, 2019; Wold et al., 2001). This feature of latent-variable methods enables them to aid process understanding solving data exploration and mining problems by interpretation of the models (Burnham et al., 2001; Camacho et al., 2010; Ergon, 2004; Kosanovich et al., 1996; Kourti, 2019; Kvalheim, 2010; Vitale et al., 2021; Wold et al., 1987a, 2001).

Process understanding is the first, fundamental task that can support the operation of industrial biorefineries. The potential of latent-variable methods to this end has been widely proven in the literature. For example, Mortensen et al. (2006) developed a parallel factor analysis model of fluorescence data to produce a chemically interpretable visualization of the progress of a fed-batch process for production of enzymes, which allowed to understand the evolution in time of the compounds of interest; they also developed a PLS model to investigate the relationship between process parameters and end-of-batch enzymatic activity, and to detect the natural end-point of the batch process. In the context of biorefineries, exploratory data analytics methods are mostly used to uncover properties of different biomasses or of different pre-treatments. For

examples Fernandes et al. (2020) developed a PCA model to understand the effects of fungal fermentation as a pre-treatment of grape stalks, while Jiang et al. (2016) used PLS to investigate the effects of process parameters on carbohydrate release in thermochemical pre-treatment of macroalgal biomass. Concerning the upstream process, Guan et al. (2015) analyzed, by PCA, the properties of genetically engineered bacterial strains to produce propionic acid and identified key metabolic nodes influencing the productivity. The very same methodology was proved to be effective to investigate a particularly complex phenomenon in the downstream: membrane fouling. Maere et al. (2012) extracted informative features from the pressure profile of a membrane bioreactor; they used them in PCA to understand the fouling trends and investigate the effects of process parameters. Klimkiewicz et al. (2016) applied a PCA-based method (multilevel simultaneous component analysis with invariant patterns) to an industrial ultrafiltration downstream of a fermenter for production of enzymes, with the intention of understanding the permeate flux decline due to fouling.

Finally, a particularly relevant achievement of latent-variable models for process understanding in industrial biorefineries is represented by the study carried out by Nachtergaele et al. (2020). They considered an industrial biorefinery for the production of fatty acids from different kinds of animal fats or vegetable oils and proposed a systematic procedure for process understanding by multivariate data analytics. The effect of various feedstocks on the performance of the hydrolysis process was investigated by PCA, while PLS was employed to understand the effect of parameters of the distillation units on the product quality and its variability.

1.2.3 Process monitoring and soft sensing

The concept of process monitoring has been introduced by Shewart (1931) in an univariate fashion. Given the increasing availability and redundancy of process data, methods to handle highly multivariate cases have been developed, proposed, and reviewed in the literature (Chiang et al., 2001; Das et al., 2012; He et al., 2018; Kourti et al., 1995; Kresta et al., 1991; MacGregor et al., 1995; Qin, 2003; Reis et al., 2017). In general: «The goal of process monitoring is to ensure the success of the planned operations by recognizing anomalies of the behavior» (Chiang et al., 2001). When process monitoring is implemented by checking the conformity of new data to the distribution of data from normal operating conditions (NOC) by means of data-driven models, it is referred to as statistical process control (SPC; MacGregor et al., 1995).

SPC can be framed in three categories according to the aim of monitoring (Kourti, 2003):

- quality monitoring focuses on the variables quantifying the product quality alone, and can be achieved by means of traditional methods applicable to single or few variables (Jackson, 1959; Montgomery, 2009);
- “general” process monitoring focuses on variables characterizing the process and follows the objective stated above, hence it can be implemented by means of latent-variable models, such as PCA (Nomikos, 1996; Wise et al., 1996);

- quality-relevant monitoring aims at assessing both the product quality and the process variables most related to the quality itself, thus it can be implemented, for example, by PLS (Li et al., 2011; Nomikos et al., 1995b).

These three categories of process monitoring can be further subdivided according to the way the objective is pursued. For example, if quality-relevant monitoring is to be implemented just to discriminate between on- and off-specification products, the problem can be solved by classification models such as PLS discriminant analysis (PLSDA) (Ballabio et al., 2013; Barker et al., 2003; Lee et al., 2018). In the context of general process monitoring, the multiclass extension of this method (Pomerantsev et al., 2018) can solve problems where one is interested in distinguishing NOC data from a set of known process faults. On the other hand, if one is interested in recognizing only NOC data, discriminating them from any, potentially unknown faulty condition (a common occurrence in the process industry), one-class modeling techniques, such as one-class PLS (Xu et al., 2011, 2013) or PLS density modeling (Oliveri et al., 2014), offer promising solutions to the problem.

Most of the methods mentioned for process monitoring require access to measurement of the product quality, some with the same frequency of the process variables. However, this simple requirement might be impossible to satisfy. For example, variables typically used to characterize the quality of fermentation/bio-based processes can be very complex in nature or require complicated and lengthy, labor-intensive laboratory analyses to be measured (Kroll et al., 2017; O'Flaherty et al., 2020). Some variables may not be directly measurable at all, such as the hydraulic resistance of a membrane quantifying its fouling state (Huang et al., 2021; Shi et al., 2014). Cases like these can be handled by exploiting the relationship between process variables and product quality (or, more generally, between easy-to-measure variables and hard-to-measure variables), using online measurements of the former to predict the latter: this concept is known as soft sensing (Camacho et al., 2008b; Kadlec et al., 2009, 2011; Lin et al., 2009; Luttmann et al., 2012; Perera et al., 2023; Souza et al., 2016; Zhou et al., 2021; Zhu et al., 2020). Soft sensors are strictly related to the problem of quality-relevant monitoring (Mainka et al., 2019), therefore the two topics are discussed together in this Section.

Chiang et al. (2006) monitored an industrial fermentation process comparing various approaches, among which PCA, a PLS model predicting a "batch maturity index", and the Tucker3 tensor decomposition. The methods enabled the identification and troubleshooting of recurrent issues with that specific fermentation, which were solved by generating a significant value for the company participating in the study. Lennox et al. (2001) considered a similar problem, exploiting PCA to monitor an industrial fed-batch fermenter in the context of general process monitoring, and PLS for quality-relevant monitoring. Sá et al. (2017) investigated several approaches to process monitoring in a microalgae membrane harvesting process. Viability and concentration of microalgae can be inferred from two-dimensional fluorescence spectra; the significant principal components of spectra (obtained by PCA) were used as inputs

to develop a PLS-based soft sensor to predict cell viability and concentration. The soft sensor was then used for quality-relevant monitoring. A PLSDA classifier was also tuned to discriminate between disrupted and viable cells. A PLS soft sensor was developed by Pontius et al. (2020) to predict the concentration of six species of interest from spectral data of a yeast fermentation process, aiming at quality-relevant monitoring.

Some studies focused on the pure predictive power of the soft sensors they developed. For example, Philippe et al. (2013) considered an array of four parallel membrane bioreactors for industrial waste-water treatment and developed a PLS soft sensor for each one of them. They aimed at predicting the dynamic evolution of membrane resistances, due to fouling, using process variables and offline measurements of the feed composition. Despite the reactors being identical, the performances of the soft sensors were sometimes not satisfactory. The authors investigated the issues and gave precious guidelines for handling predictive models in such complex scenarios. A similar problem was tackled by Kaneko et al. (2013), who developed two PLS soft sensors to estimate the evolution of trans-membrane pressures (TMPs) of two industrial membrane bioreactors. They noted that predicting the TMP rather than resistances yields good performance.

In the context of biorefinery, the study of Holm-Nielsen et al. (2011) represents a relevant contribution. They considered a pilot scale biorefinery processing manure for biogas production by anaerobic digestion. Multiple PLS soft sensors were calibrated to predict the total concentration of volatile fatty acids and concentrations of specific acids of interest starting from near-infrared spectra of the off-gas. The models yielded excellent performance and served as basis to develop a quality-relevant monitoring system to aid process operation by prompt identification of faults.

1.2.4 Data-driven process improvement and optimization

The use of latent-variable models to develop process understanding, monitoring systems, and soft sensors is the foundation for data-driven process improvement and optimization. Soft sensors, as commonly intended in the process industry, allow for the estimation of the product quality obtained from given process conditions. One might therefore think to use the soft sensor "the other way around", setting a target quality and computing process conditions that, according to the model, allow to manufacture a product with that assigned quality. This is the problem tackled by latent-variable model inversion (LVMI; Arce et al., 2021; Jaeckle et al., 1998, 2000; Ruiz et al., 2018; Tomba et al., 2012a, 2013b). While this task could be accomplished by any optimization method applied to any process model relating the product quality to process variables (possibly both nonlinear), LVMI offers a remarkable advantage: if models are derived on historical process data, only the input and output spaces covered by the data are considered as feasible, according to the models. While this could appear as a limitation if one wants to design a product significantly different from the ones in the historical database,

it offers the advantage that process constraints and production policies are implicitly encapsulated in the model, hence the result of the LVMI procedure will automatically comply with such constraints (Jaeckle et al., 1998). This feature of latent-variable models is of paramount importance in data-driven process improvement. Models exclusively based on data rely on pure correlation rather than causality (Reis et al., 2019), thus care must be taken when they are used for sensitive tasks, as product design. However, latent-variable models can extract meaningful information from daily operation data and incorporate it in the space of latent variables, implicitly constraining the set of possible solution of data-driven optimization problems, such as LVMI, to respect the process structure, thus offering guarantees on its feasibility and effectiveness (Ferrer, 2021).

One may want to make a step further: what if a product is already being manufactured in a plant and needs to be produced in a second, similar plant, possibly at a different scale? Joint-Y partial least-squares (JYPLS) regression has been proposed as a variation of PLS and combined with LVMI to tackle this kind of problem (García Muñoz, 2014; García-Muñoz, 2004; García-Muñoz et al., 2005). Considering biorefineries based on the biological conversion platform, such as fermentation, the problem of scale-up is particularly relevant (Woodley, 2020) as it is known that biological processes suffer from strong scale effects (Burgard et al., 2016; Facco et al., 2020). Data analytics has been deemed to have a great potential to this end (Cuellar et al., 2020). Another critical step of biorefinery development, the selection of the biomass feedstock (Attard et al., 2020), could draw significant advantages from the very same approach.

However, LVMI is still a relatively young approach and found limited to no applications in the domain of bio-based processes. Research on LVMI mostly concerns the pharmaceutical and industrial chemistry sectors. For example, Liu et al. (2011a) exploited PLS model inversion (the most widespread LVMI approach) to define the optimal formulation of a pharmaceutical product to be produced in an industrial tablet manufacturing line, and also identified the best process conditions to this end. Tomba et al. (2013b) designed process conditions for quality improvement in a wet granulation operation, discussing cases where not all quality variables to describe the product have a specified target. In another significant application, Jaeckle et al. (2000) identified, by PLS model inversion, the process conditions of an industrial fed-batch emulsion polymerization reactor for consistent quality operation. García-Muñoz et al. (2006) used LVMI to design reference trajectories for the profiles of variables in an industrial pulp digester for paper production, operating in batch regime.

In a similar application to the latter, García-Muñoz et al. (2005) exploited LVMI and JYPLS to scale-up a pulp digester, from pilot to industrial scale. Tomba et al. (2014) considered a nanoparticle production process, for which they developed a JYPLS model and inverted it to transfer a product of a small-scale unit to a pilot-scale unit. One particularly relevant application is the by study by Dal-Pastro et al. (2017). They considered the product transfer problem in a wheat milling process. A new variety of wheat (feedstock) needed to be processed at industrial-

scale to yield a given product quality; however, limited data on that process/feedstock combination were available for the industrial unit, while a large dataset was available for a laboratory-scale unit. JYPLS model inversion successfully identified process conditions to achieve the goal. In the context of biological processes, Facco et al. (2020) used a JYPLS model inversion to aid the selection of mammalian cell lines for biopharmaceutical production while aiding the scale-up process. They considered six scales (from static wells to small scale shaken flasks), proving that JYPLS can provide precious information on common variability drivers across scales and point out specific differences across scales. They also showed how JYPLS modeling can efficiently use information on small, data-rich scales to infer important information on larger scales with smaller datasets, as also proved by Dal-Pastro et al. (2017).
Finally, JYPLS has been studied to transfer not only the product across plants and scales, but the models themselves. Transfer of process monitoring models was achieved on both continuous systems (Facco et al., 2012) and batch systems (Facco et al., 2014), possibly also integrating prior, mechanistic knowledge on the process (Tomba et al., 2012b). Strategies to speed up model transfer by data culling have been proposed as well (Chu et al., 2021).

1.3 Research objectives

In the past decades, data-driven modeling proved to be an effective tool to support operation, improvement, and optimization of industrial processes, as testified by the large number of studies focusing on these tasks. However, as discussed in Section 1.2.1, these methods are still not widely applied to industrial biorefineries, partly due to the limited number of industrial-scale biorefineries, partly because heavily nonlinear, high-complexity approaches promising pure performance (at the expense of interpretability and, sometimes, robustness) seem to be the default choice of practitioners and researchers. The second point is also related to the plethora of data-driven models available and to the overwhelming task of choosing the most appropriate one, which usually leads to the comparison of a very limited number of models, the ones the analyst is most accustomed with. While one could argue that this approach is simplistic and claim that every model relies on some assumptions that can be verified beforehand, often there is no established criterion to assess the "simple" proprieties models rely on (such as presence of nonlinear relationships) considering data alone, especially in highly multivariate scenarios.
In light of these points, the main scientific contribution this Thesis aspires to give can be framed as follows.

1. **Provide evidence that Industry 4.0 is a precious tool for industrial biorefineries.**

This objective is accomplished by applying data analytics techniques to support the operation of the world's first plant to produce BDO from renewable biomass, either by data-driven process improvement, for example process understanding or product design, or by developing model-based support systems, such as monitoring systems or soft sensors. The modeling

methodologies are selected by a rational reasoning: find the simplest model (according to some complexity scale) that can satisfactorily accomplish the task of interest (Rendall et al., 2019). Particular attention is devoted to problems typical of biorefineries and biological processes, such as the presence of multiple parallel units manufacturing the same product in the upstream (bioconversion), or issues causing processes to run in semi-continuous mode in the downstream (membrane filtration processes). It is important to note that, while the nature of the aforementioned tasks is mostly procedural and geared towards application of data-driven modeling, they have a remarkable research value nonetheless due to the industrial environment they are applied to, implementing a one-of-a-kind, highly innovative process.

It is reasonable to conjecture that specific issues of existing methods will be found while pursuing the objective stated above, which could disclose paths for improvement of data-driven modeling. Therefore, the second fundamental objective of this Thesis can be states as follows.

2. **Contribute to the methodological advancement of data-driven modeling.**

This objective is accomplished by pursuing the opportunities for improvement uncovered in the application of existing techniques, or facing problems for which no existing method is fully appropriate. A particularly ambitious objective is to develop guidelines for selecting the most appropriate model for a specific task, say process monitoring. To this end, methods to check inherent characteristics of the data at hand must be developed as to guide the selection of the best candidate models (based on the assumptions they rely on) in a set of relevant models. This is a new approach to model discrimination, as opposed to the common "winner takes all" rationale based on comparison of the validation errors of a large pool of models. Some work has been done already on this topic, even though restricted to regression models and focusing on relationship between a single output variable and a limited number of input variables (Sun et al., 2021).

A common rationale underly both the objectives described above: the application of data-driven modeling can be significantly aided by the incorporation of domain-specific knowledge, whether on the process generating the data or on the properties of the data themselves. This principle serves as a guiding light in all the studies presented in this Thesis, its nature being methodological rather than aimed at the solution of a specific problem or at the improvement of a specific method. In fact, the word "improve" may be interpreted in broad sense, for example as to improve the performance yielded by the models for a given objective, or to improve the selection process of the best model for a given application, as described above. However, a particularly significant interpretation is to blend highly abstracted information concealed in process data with mechanistic knowledge of the physics/chemistry/biology of the process. Among the multiple possibilities to do so, two will be prioritized in this Thesis:

- the combination of data-driven models and knowledge-based models according to the **hybrid modeling** paradigm (Narayanan et al., 2023; Rajulapati et al., 2022; Sansana et al., 2021; Solle et al., 2017; von Stosch et al., 2014; Yang et al., 2020);

- the use of process knowledge to extract relevant information from data by synthesizing informative features, for example by "compression" of the time profiles of variables in scalar variables (or by augmentation of the data already available computing additional variables) according to the **feature-oriented modeling** paradigm (Reis et al., 2022; Rendall et al., 2019; Yoon et al., 2001).

1.4 Thesis roadmap

The research discussed in this Thesis is organized according to the two research objectives outlined in the previous Section. A scheme of the main topics discussed herein is reported in Figure 1.6. Chapter 2 sets the mathematical background of the Thesis. Support systems for the industrial biorefinery are discussed in Chapter 3, Chapter 4, and Chapter 5. Finally, Chapter 6 and Chapter 7 present contributions to the advancement of data-driven modeling.

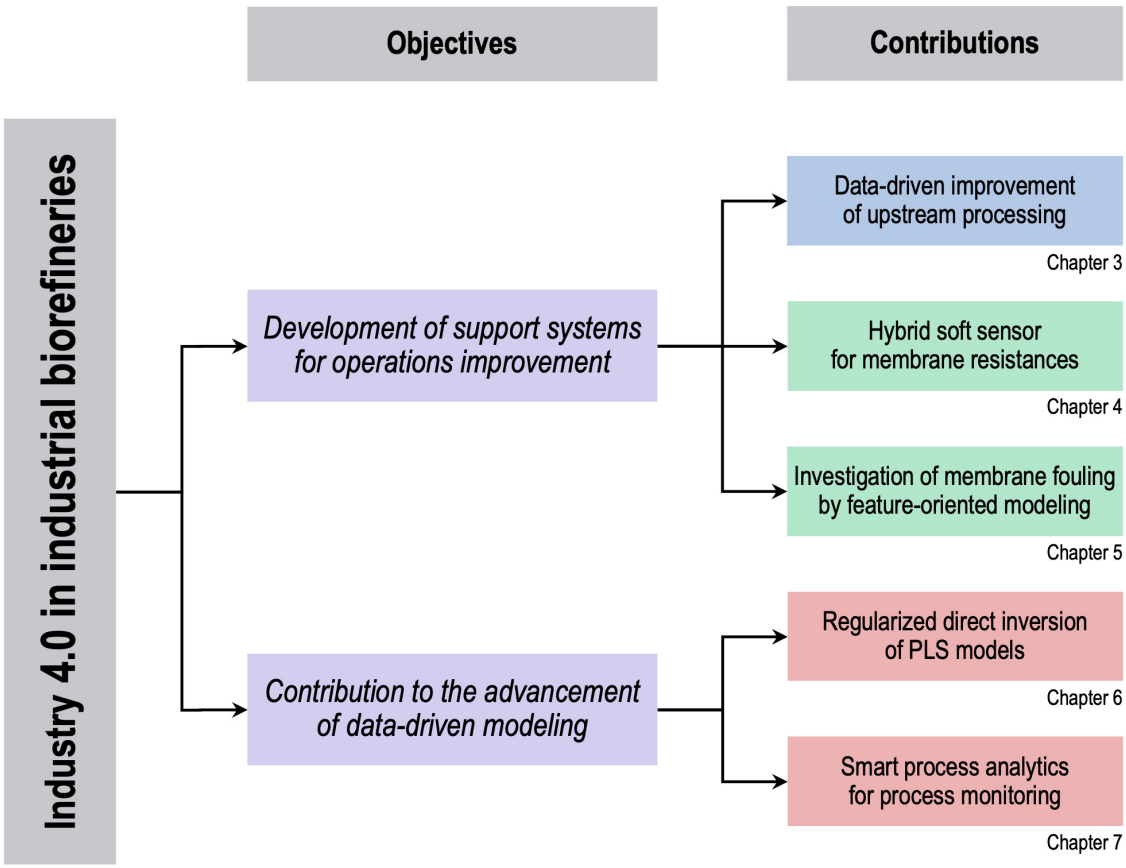


Figure 1.6. *Graphical representation of the Thesis roadmap.*

Chapter 2 describes the fundamental mathematical methods leveraged in the studies reported in this Thesis. However, some of the methods are highly specific only to the study discussed in Chapter 7 and require a strong contextualization into the problem (fault detection), therefore they are introduced in Chapter 7 and not in Chapter 2. The mathematical symbols are consistent

throughout this Thesis and are defined upon their first use, together with the dimensions of vectors and matrices, and the units of measurement of physical quantities (when relevant).

Chapter 3 describes the data-driven improvement of the bioconversion step of the upstream process, where seven fed-batch bioreactors are operated in parallel. A decreasing trend in the end-of-batch product quality developed over several months of operations, affecting all the bioreactors. The properties of latent-variable models are leveraged to troubleshoot the issue. PCA is first used to investigate potential differences among the bioreactors and to develop process understanding. A JYPLS model of the end-of-batch product quality is then developed and interpreted to gain insight on the causes of the quality loss. Finally, a multivariate approach based on LVMI is adopted to develop quantitative guidelines to recover the product quality.

Chapter 4 discusses the implementation of a support system to enhance a membrane filtration unit in the downstream section. Seven interconnected membrane modules realize an ultrafiltration of the outlet of bioreactors to separate biomass from the solution containing the product of the bioconversion. The equipment suffers from membrane fouling issues due to the nature of the feed. However, the fouling monitoring system adopted in the process relies on measurement of operator-read instrumentation installed on the plant, which are plagued by high variability and influenced by process changes, hindering the interpretation of the fouling trends of individual membranes. A soft sensor for online estimation of the resistances of all membrane modules is developed to improve process operation. The soft sensor is based on a hybrid model: first, a PLS model estimates trans-membrane pressures of all the membrane modules; then, the outputs of the PLS model are used to compute the resistances by a physics-based model, that is Darcy's law. The estimated resistances are available in real time and enable the effective monitoring of both reversible fouling (caused by the deposition of material on the membranes) and irreversible fouling (causing membrane degradation over time).

Chapter 5 outlines a comprehensive investigation of membrane fouling in the same process considered in the previous Chapter. The investigation is carried out by feature-oriented modeling. This method allows to elegantly solve issues induced by membrane fouling preventing the application of standard data analytics methods (for example the high variability of the duration of filtration batches) while simultaneously building process knowledge into the data analytics workflow by enhancing the information on the phenomena of interest (membrane fouling). Numerous process settings potentially related to fouling are considered in the analysis. A general screening procedure is proposed to cope with the large number of features and to identify the process settings most closely related to membrane fouling. The analysis confirms the effectiveness of the cleaning policies adopted by the plant and uncovers a strong interaction between reversible and irreversible fouling, offering precious guidelines to improve the maintenance schedule of membranes.

Chapter 6 presents a novel method for the algebraic inversion of PLS models. PLS model inversion can be achieved by algebraic manipulation of the model equations. However, the

inversion procedure requires the output variables to be independent, prescribing to remove correlated output variables before model calibration. This leads to a loss of information as some quality variables are not considered in modeling, therefore they could not comply with the product quality specification upon implementation of the inversion solution. An improved formulation of PLS model inversion is proposed. The algorithm allows to tackle the output correlation by design: the information provided by all output variables is retained in the modeling step, while the non-systematic part is removed in the inversion step by regularization. The advantages of the proposed approach are demonstrated on two simulated fermentation processes.

Chapter 7 introduces a framework for the automatic selection and calibration of data-driven models for fault detection geared towards industrial manufacturing processes. Only data from normal operating conditions are required by the framework, and no prior assumption on the fault modes of the process is made. A preliminary data interrogation is conducted to assess characteristics of the data relevant to model selection, such as presence of nonlinear correlation among variables (equivalent to non-normality of the distribution of data), presence of dynamics in the data, and availability of variables describing the product quality. A subset of candidate models able to cope with the found data properties is selected from models included in the library provided with the framework. Finally, the best candidate is identified by a rigorous model selection procedure. The effectiveness of the framework is tested of four case studies: a simulated linear, static dataset; the Tennessee Eastman Process simulator; a simulation of a process for continuous filtering and drying of paracetamol; an industrial dataset from a metal etching process for semiconductor manufacturing. The framework identifies the most appropriate model (among the ones included in the library) for fault detection in all cases, as proved by the fault detection performance on data from faulty conditions not used for calibration.

Chapter 2

Mathematical background

This Chapter introduces the fundamental mathematical methods used throughout this Thesis. The family of latent-variable models is of particular interest: it includes PCA, PLS, and canonical variate analysis (CVA). These methods have a long record of successful applications in (bio-)chemical engineering and are described in this Chapter. Advanced methods meant to handle parallel units, design process conditions to achieve a given target, and deal with data from batch processes are introduced as well. Finally, fundamental concepts of hybrid modeling are discussed. Note that some mathematical concepts will be introduced in Chapter 7 rather than in this Chapter, with a strong contextualization into the problem of interest: fault detection in manufacturing processes.

2.1 Principal component analysis (PCA)

2.1.1 Model calibration

PCA (Hotelling, 1933; Pearson, 1901; Wold et al., 1987a) is a multivariate data analytics method aimed at dimensionality reduction of a data matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$ containing N observations (rows) of V_X variables (columns). The method extracts a sequence of A independent variables, called principal components (PCs), formulated as linear combinations of the original variables in **X**. Provided that matrix **X** is autoscaled, meaning its columns are mean-centered and scaled to unit variance, the PCA model is defined as a matrix decomposition:

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{P}^{\mathrm{T}} + \mathbf{E} \quad , \tag{2.1}$$

where:

- $\mathbf{T} \in \mathbb{R}^N \times \mathbb{R}^A$ is the score matrix, the columns of which represent the PCs and rows are the projections of the observations in **X** onto the space of PCs;
- $\mathbf{P} \in \mathbb{R}^{V_X} \times \mathbb{R}^A$ is the loading matrix, which contains coefficients to formulate the PCs as linear combinations of the original variables (note that columns of **P** are independent and scaled to unit norm);
- $\mathbf{E} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$ is the residual matrix, containing the portion of **X** not modeled by PCA.

In (2.1), · represents the row-by-column matrix product and the superscript T denotes the matrix transposition operation. A schematic representation of the PCA model is reported in Figure 2.1.

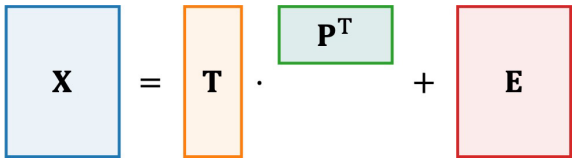


Figure 2.1. *Schematic representation of the PCA model. Matrices involved in the decomposition in (2.1) are represented as rectangles.*

The PCs are computed sequentially as to maximize the variance captured by each one of them, and to be orthogonal to each other. Therefore, the PCA model can be calibrated by applying singular-value decomposition (SVD; Golub et al., 2013). Said $R_X = \text{rank}(\mathbf{X}) \leq \min\{N, V_X\}$, the SVD of $\mathbf{X}$ is:

$$\mathbf{X} = \mathbf{N} \cdot \mathbf{\Sigma} \cdot \mathbf{O}^{\mathrm{T}} = [\mathbf{N}_1 \quad \mathbf{N}_2] \cdot \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \cdot [\mathbf{O}_1 \quad \mathbf{O}_2]^{\mathrm{T}} \quad , \tag{2.2}$$

where $\mathbf{N} \in \mathbb{R}^N \times \mathbb{R}^{R_X}$ and $\mathbf{O} \in \mathbb{R}^{V_X} \times \mathbb{R}^{R_X}$ are orthonormal matrices the columns of which, called left and right singular vectors, respectively, are the bases of the row space and column space of $\mathbf{X}$, respectively, while $\mathbf{\Sigma} \in \mathbb{R}^{R_X} \times \mathbb{R}^{R_X}$ is a diagonal matrix containing the singular values of $\mathbf{X}$ sorted from the largest to the smallest on the diagonal. The SVD allows for a perfect reconstruction of matrix $\mathbf{X}$; however, setting a value $A < R_X$, one can truncate the SVD to obtain the best rank-A reconstruction of $\mathbf{X}$, which is equivalent to obtain the PCA model of $\mathbf{X}$ with A PCs. This is done splitting matrix $\mathbf{N}$ into $\mathbf{N}_1 \in \mathbb{R}^N \times \mathbb{R}^A$ and $\mathbf{N}_2 \in \mathbb{R}^N \times \mathbb{R}^{R_X - A}$, $\mathbf{O}$ into $\mathbf{O}_1 \in \mathbb{R}^{V_X} \times \mathbb{R}^A$ and $\mathbf{O}_2 \in \mathbb{R}^{V_X} \times \mathbb{R}^{R_X - A}$, and $\mathbf{\Sigma}$ into $\mathbf{\Sigma}_1 \in \mathbb{R}^A \times \mathbb{R}^A$ and $\mathbf{\Sigma}_2 \in \mathbb{R}^{R_X - A} \times \mathbb{R}^{R_X - A}$. The matrices of the PCA model (2.1) are then:

$$\mathbf{T} = \mathbf{N}_1 \cdot \mathbf{\Sigma}_1 \quad , \tag{2.3}$$

$$\mathbf{P} = \mathbf{O}_1 \quad , \tag{2.4}$$

$$\mathbf{E} = \mathbf{N}_2 \cdot \mathbf{\Sigma}_2 \cdot \mathbf{O}_2^{\mathrm{T}} \quad . \tag{2.5}$$

Figure 2.2 shows a geometric interpretation of the PCA model.

The value of A is set to include in $\mathbf{\Sigma}_1$ only the significant singular values. Several approaches exists to assess the significance of singular values: interested readers are referred to relevant literature resources (Bro et al., 2008; Camacho et al., 2012, 2014; Eastment et al., 1982; Jackson, 1991; Louwerse et al., 1999a; Saccenti et al., 2015a, 2015b; Valle et al., 1999; Vitale et al., 2017; Wold, 1978; Zwick et al., 1986). Given the relationship between PCA and SVD, scores describe the relationship among observations, while loadings allow to understand the relationship among variables (Kosanovich et al., 1996; Wold et al., 1987a). More sophisticated tools for PCA model interpretation exist nonetheless, such as the structural and variance information plot (Camacho et al., 2010).

2.1.2 Model application

Once the PCA model has been calibrated, new observations can be projected onto the space of PCs exploiting (2.1). Given a new observation $\mathbf{x}_{\text{new}} \in \mathbb{R}^{V_X}$ (the components of which are assumed to be scaled with the means and standard deviations of the columns of the calibration

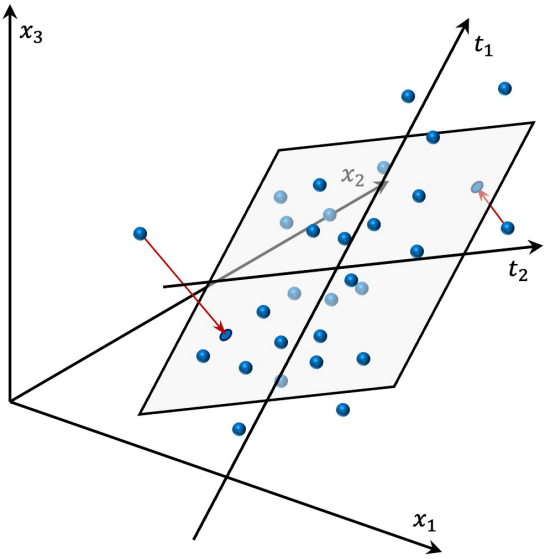


Figure 2.2. *Geometric interpretation of the PCA model with A = 2 PCs (t_1 and t_2) derived from a data matrix containing V_X = 3 variables (x_1, x_2, and x_3). The PCA model identifies a subspace of the space of input variables defining the new coordinate systems (the PCs) as the directions of the maximum variability in the original space, with orthogonality constraints among all the PCs.*

matrix $\mathbf{X}$), its projection, $\mathbf{t}_{\text{new}} \in \mathbb{R}^A$, can be computed as:

$$\mathbf{t}_{\text{new}}^{\text{T}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{P} \quad . \tag{2.6}$$

The best rank-A reconstruction of $\mathbf{x}_{\text{new}}$ can then be computed using (2.1):

$$\hat{\mathbf{x}}_{\text{new}}^{\text{T}} = \mathbf{t}_{\text{new}}^{\text{T}} \cdot \mathbf{P}^{\text{T}} \quad , \tag{2.7}$$

and the reconstruction residuals can be defined as:

$$\mathbf{e}_{\text{new}} = \mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}} = \left(\mathbf{I}_{V_X} - \mathbf{P} \cdot \mathbf{P}^{\text{T}}\right) \cdot \mathbf{x}_{\text{new}} \quad , \tag{2.8}$$

where $\mathbf{I}_{V_X} \in \mathbb{R}^{V_X} \times \mathbb{R}^{V_X}$ is the identity matrix.

2.1.3 Prediction diagnostics

The reliability of the reconstruction of a new observation by PCA model application can be measured by means of two diagnostic statistics (Wise et al., 1996):

- the T_X^2 statistic (Jackson, 1959) measures the squared distance of the projection of a new observation from the center of the PC space:

  $$T_X^2 = \mathbf{t}_{\text{new}}^{\text{T}} \cdot \mathbf{\Lambda}_T^{-1} \cdot \mathbf{t}_{\text{new}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{P} \cdot \mathbf{\Lambda}_T^{-1} \cdot \mathbf{P}^{\text{T}} \cdot \mathbf{x}_{\text{new}} \quad , \tag{2.9}$$

  where $\mathbf{\Lambda}_T \in \mathbb{R}^A \times \mathbb{R}^A$ is a diagonal matrix containing the eigenvalues of $\mathbf{X}$ (squared singular values) from the calibration dataset:

  $$\mathbf{\Lambda}_T = \mathbf{T}^{\text{T}} \cdot \mathbf{T} = \mathbf{\Sigma}_1^{\text{T}} \cdot \mathbf{\Sigma}_1 \quad ; \tag{2.10}$$

- the Q_X statistic (Jackson et al., 1979), also referred to as squared prediction error, measures the squared orthogonal distance between a new observation and the space of PCs:

  $$Q_X = \mathbf{e}_{\text{new}}^{\text{T}} \cdot \mathbf{e}_{\text{new}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \left(\mathbf{I}_{V_X} - \mathbf{P} \cdot \mathbf{P}^{\text{T}}\right) \cdot \mathbf{x}_{\text{new}} \quad . \tag{2.11}$$

Note that, in this Thesis, the Q_X statistic is assumed to be computed on a pre-processed observation $\mathbf{x}_{\text{new}}$,which allows to equally weight all variables to avoid uneven contributions due to variables having different magnitudes and scales of variability (Fernandes et al., 2022).
The significance of the T_X^2 and Q_X statistics can be assessed comparing their values to confidence limits derived from their distributions. The confidence limit for the T_X^2 can be estimated on the basis of the F distribution (Jackson, 1959), the β distribution (Tracy et al., 1992), or the χ^2 distribution with matching moments (Nomikos et al., 1995a). On the other hand, the confidence limit of the Q_X statistic can be obtained using a weighted summation of χ^2 distributions (Box, 1954), a simple approximation of such distribution (Jackson et al., 1979), or the χ^2 distribution with matching moments (Nomikos et al., 1995a). More sophisticated methods free from distributional assumptions, for example based on kernel density estimation, have been proposed as well (Martin et al., 1996). A complete description of all the approaches to estimate the confidence limits of T_X^2 and Q_X can be found in the literature (Qin, 2003; Reis et al., 2021a; Thissen et al., 2001; Tracy et al., 1992), and details on some relevant methods to are discussed in Section 7.2.2.
Note that the T_X^2 and Q_X statistics can also be computed for each observation in the calibration dataset using equations equivalent to (2.9) and (2.11). This is helpful, for example, in diagnosing anomalous observations that could bias the model. Such a feature is the basis for process monitoring by PCA (Kourti et al., 1995, 1996; Nomikos et al., 1994, 1995a; Qin, 2003; Wise et al., 1996) as discussed in Section 1.2.3. This feature of PCA will be further discussed in Section 7.2.1.

2.2 Partial least-square (PLS) regression

2.2.1 Model calibration

Given a matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$, gathering N obersvations of V_X input (predictor) variables, and a matrix $\mathbf{Y} \in \mathbb{R}^N \times \mathbb{R}^{V_Y}$, containing the same number of observations of V_Y output (predicted) variables, PLS (Geladi et al., 1986; Wold, 1966; Wold et al., 2001) provides models for both data matrices together with a linear regression model between them. This is done by identifying two sequences of A mutually orthogonal latent variables (LVs), one for the inputs and one for the outputs, defined as linear combinations of the input and output variables, respectively. If both matrices are autoscaled, the data models are provided as:

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{P}^{\mathrm{T}} + \mathbf{E} \quad , \tag{2.12}$$

$$\mathbf{Y} = \mathbf{U} \cdot \mathbf{Q}^{\mathrm{T}} + \mathbf{F} \quad , \tag{2.13}$$

where:

- $\mathbf{T} \in \mathbb{R}^N \times \mathbb{R}^A$ is the input score matrix, the columns of which represent the input LVs and rows are the projections of the observations in $\mathbf{X}$ onto the space of input LVs;

- $\mathbf{P} \in \mathbb{R}^{V_X} \times \mathbb{R}^{A}$ is the input loading matrix, which contains coefficients to formulate the matrix decomposition model of the input variables;
- $\mathbf{E} \in \mathbb{R}^{N} \times \mathbb{R}^{V_X}$ is the residual matrix of the data model for $\mathbf{X}$, containing the input "reconstruction" residuals;
- $\mathbf{U} \in \mathbb{R}^{N} \times \mathbb{R}^{A}$ is the output score matrix, the columns of which represent the output LVs and rows are the projections of the observations in $\mathbf{Y}$ onto the space of output LVs;
- $\mathbf{Q} \in \mathbb{R}^{V_Y} \times \mathbb{R}^{A}$ is the output loading matrix, which contains coefficients to formulate the matrix decomposition model of the output variables;
- $\mathbf{F} \in \mathbb{R}^{N} \times \mathbb{R}^{V_Y}$ is the residual matrix of the data model for $\mathbf{Y}$, containing the output "reconstruction" residuals.

However, the PLS model is not made by two mere PCA models of $\mathbf{X}$ and $\mathbf{Y}$. In fact, LVs are defined to maximize the modeled cross-covariance between $\mathbf{X}$ and $\mathbf{Y}$ (therefore between pairs of input and output LVs), and, at the same time, the variance of data matrices modeled by each LV in the data models. The first objective is accomplished defining a weight matrix $\mathbf{W} \in \mathbb{R}^{V_X} \times \mathbb{R}^{A}$, computed one column at a time. For example, this first one is the eigenvector of the cross-covariance $\mathbf{X}^{\mathrm{T}} \cdot \mathbf{Y} \cdot \mathbf{Y}^{\mathrm{T}} \cdot \mathbf{X}$ corresponding to its largest eigenvalue. The second objective is accomplished defining a matrix of adjusted (or rotated) weights, $\mathbf{W}^{*} \in \mathbb{R}^{V_X} \times \mathbb{R}^{A}$, as:

$$\mathbf{W}^{*} = \mathbf{W} \cdot (\mathbf{P}^{\mathrm{T}} \cdot \mathbf{W})^{-1} \quad . \tag{2.14}$$

Adjusted weights are then used to project input observations onto the space of LVs:

$$\mathbf{T} = \mathbf{X} \cdot \mathbf{W}^{*} \quad . \tag{2.15}$$

This results in the maximization of linear correlation between pairs of input and output LVs. In fact, the $\mathbf{X}$-to-$\mathbf{Y}$ regression model is provided as a sequence of A additive linear regression models between corresponding pairs of LVs, represented in compact notation as:

$$\mathbf{U} = \mathbf{T} \cdot \mathrm{diag}(\mathbf{b}) + \mathbf{V} \quad , \tag{2.16}$$

where vector $\mathbf{b} \in \mathbb{R}^{A}$, called inner regression coefficients, lays on the diagonal of the square matrix $\mathrm{diag}(\mathbf{b})$, and $\mathbf{V} \in \mathbb{R}^{N} \times \mathbb{R}^{A}$ is the matrix of inner regression residuals. A schematic representation of the PLS model is reported in Figure 2.3.

The number of LVs, A, is usually set as to maximize the predictive performance of the PLS model on data not used for calibration. Cross-validation is a widely used approach (Bro et al., 2008; Geladi et al., 1986; Louwerse et al., 1999a; Wold et al., 2001); other approaches, such as parameter population analysis (Deng et al., 2015) or information criteria (Krämer et al., 2011), exist nonetheless. Once A has been set, all the entities in the PLS model ($\mathbf{T}$, $\mathbf{P}$, $\mathbf{U}$, $\mathbf{Q}$, $\mathbf{W}$, and $\mathbf{b}$) can be computed by any PLS calibration algorithm, such as the nonlinear iterative partial least-squares algorithm (Geladi et al., 1986; Wold, 1966; Wold et al., 2001) or the statistically-inspired modification of PLS algorithm (de Jong, 1993). Additional methods exist: interested readers are referred to literature resources (Andersson, 2009; Burnham et al., 1996, 1999; Hoskuldsson, 1988). Similarly to PCA, the analysis of PLS model entities allows for data interpretation (Burnham et al., 2001).

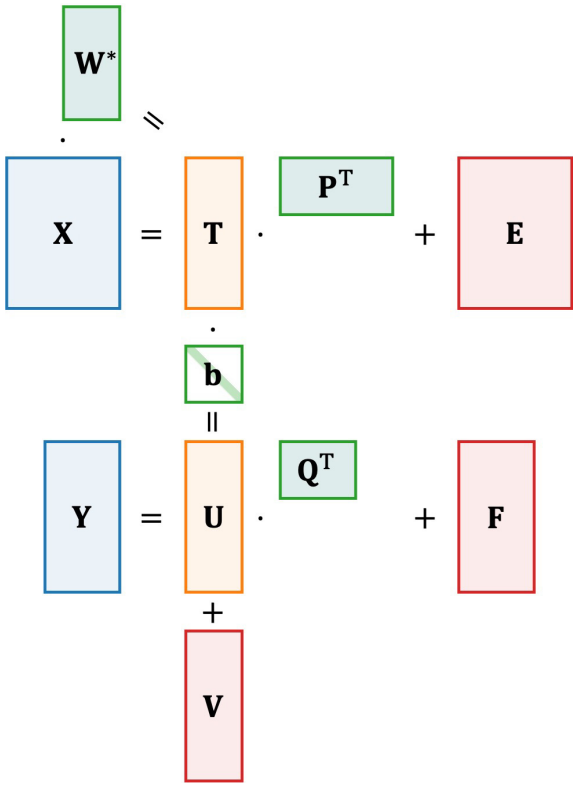


Figure 2.3. *Schematic representation of the PLS model. The horizontal equations represent the data models provided by PLS in (2.12) and (2.13). The vertical equation represents the regression model composed of (2.14), (2.15), and (2.16).*

2.2.2 Model application

Given a new input observation $\mathbf{x}_{\text{new}} \in \mathbb{R}^{V_X}$ (assumed to be scaled in the same way as the input matrix $\mathbf{X}$), the calibrated PLS model can be used to compute $\hat{\mathbf{y}}_{\text{new}}$, an approximation of the true, unknown output observation $\mathbf{y}_{\text{new}} \in \mathbb{R}^{V_Y}$ (scaled as well). The first step is to project $\mathbf{x}_{\text{new}}$ onto the space of input LVs:

$$\mathbf{t}_{\text{new}}^{\text{T}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{W}^* \quad . \tag{2.17}$$

An approximation of the output scores is then computed using the inner regression model:

$$\hat{\mathbf{u}}_{\text{new}}^{\text{T}} = \mathbf{t}_{\text{new}}^{\text{T}} \cdot \text{diag}(\mathbf{b}) \quad . \tag{2.18}$$

Finally, the approximated output scores are projected back to the space of output variables by the $\mathbf{Y}$ matrix model:

$$\hat{\mathbf{y}}_{\text{new}}^{\text{T}} = \hat{\mathbf{u}}_{\text{new}}^{\text{T}} \cdot \mathbf{Q}^{\text{T}} \quad . \tag{2.19}$$

Equations (2.17), (2.18), and (2.19) can be jointed into a single equation:

$$\hat{\mathbf{y}}_{\text{new}}^{\text{T}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{W}^* \cdot \text{diag}(\mathbf{b}) \cdot \mathbf{Q}^{\text{T}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{B} \quad , \tag{2.20}$$

where $\mathbf{B} \in \mathbb{R}^{V_X} \times \mathbb{R}^{V_Y}$ is the matrix of PLS outer regression coefficients.

2.2.3 Prediction uncertainty

As $\hat{\mathbf{y}}_{\text{new}}$ is merely an approximation of the true output observation $\mathbf{y}_{\text{new}}$, the true value of which is generally unknown, considering the prediction uncertainty is of paramount importance to

assess the reliability of the estimate. In the case of PLS prediction uncertainty (Faber et al., 1997, 2002), the confidence interval (CI) of the predicted value $\hat{\mathbf{y}}_{\text{new}}$ at significance level α can be formulated as:

$$\text{CI}(\hat{\mathbf{y}}_{\text{new}}) = \hat{\mathbf{y}}_{\text{new}} \pm \mathbf{s}_{\hat{\mathbf{y}}_{\text{new}}} t|_{\frac{\alpha}{2}} \quad , \tag{2.21}$$

where $t|_{\frac{\alpha}{2}}$ is the value of a t-distributed variable evaluated at probability $\alpha/2$, with $\alpha \in [0, 1]$ (usually $\alpha = 0.95$ or $\alpha = 0.99$). The standard deviation of $\hat{\mathbf{y}}_{\text{new}}$, $\mathbf{s}_{\hat{\mathbf{y}}_{\text{new}}} \in \mathbb{R}^{V_Y}$, is estimated as:

$$\mathbf{s}_{\hat{\mathbf{y}}_{\text{new}}} = \mathbf{MSE} \sqrt{1 + \frac{1}{N} + h_{\hat{\mathbf{y}}_{\text{new}}}} \quad , \tag{2.22}$$

where $h_{\hat{\mathbf{y}}_{\text{new}}}$ is the leverage of "observation" $\mathbf{y}_{\text{new}}$ on the model:

$$h_{\hat{\mathbf{y}}_{\text{new}}} = \frac{\mathbf{t}_{\text{new}}^{\text{T}} \cdot \boldsymbol{\Lambda}_T^{-1} \cdot \mathbf{t}_{\text{new}}}{N-1} \quad , \tag{2.23}$$

and $\boldsymbol{\Lambda}_T \in \mathbb{R}^A \times \mathbb{R}^A$ is a diagonal matrix containing unscaled variances of the LVs from the calibration dataset:

$$\boldsymbol{\Lambda}_T = \mathbf{T}^{\text{T}} \cdot \mathbf{T} \quad , \tag{2.24}$$

while $\mathbf{MSE} \in \mathbb{R}^{V_Y}$ is the vector of mean-squared errors (MSE) in calibration of output variables:

$$\mathbf{MSE} = \sqrt{\frac{\sum_{n=1}^{N} (\mathbf{y}_n - \hat{\mathbf{y}}_n)^2}{N-(A+1)}} \quad . \tag{2.25}$$

In (2.25), $A + 1$ represents the degrees of freedom of the PLS model. This is the so-called naïve PLS degrees of freedom, but more sophisticated estimators exist (Krämer et al., 2011; Van Der Voet, 1999).

2.2.4 Prediction diagnostics

Data models provided by PLS can also be used to reconstruct input and output observations, which is particularly relevant for the input ones. In fact, $\mathbf{x}_{\text{new}}$ can be projected on the space of LVs by means of (2.17), while a projection $\mathbf{t}_{\text{new}}$ can be projected back to the input space by (2.12), obtaining a rank-A reconstruction $\hat{\mathbf{x}}_{\text{new}}$. Therefore, the reconstruction error can be computed as:

$$\mathbf{e}_{\text{new}} = \mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}} = \left(\mathbf{I}_{V_X} - \mathbf{P} \cdot (\mathbf{W}^*)^{\text{T}}\right) \cdot \mathbf{x}_{\text{new}} \quad . \tag{2.26}$$

This feature of PLS can be used to define diagnostics, similar to the ones from PCA, to identify anomalous observations in the calibration dataset or to assess the reliability of the PLS model application to new observations. Such statistics are generally defined on the space of input LVs due to it being representative of the input-output covariance (Kourti, 2003; Kourti et al., 1995, 1996; Nomikos et al., 1995b):

- the T_X^2 statistics measures the squared distance of the projection of a new input observation from the center of the input LV space:

$$T_X^2 = \mathbf{t}_{\text{new}}^{\text{T}} \cdot \boldsymbol{\Lambda}_T^{-1} \cdot \mathbf{t}_{\text{new}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{W}^* \cdot \boldsymbol{\Lambda}_T^{-1} \cdot (\boldsymbol{W}^*)^{\text{T}} \cdot \mathbf{x}_{\text{new}} \; ; \tag{2.27}$$

- the Q_X statistics measures the squared orthogonal distance between a new input observation and the input space of LVs:

$$Q_X = \mathbf{e}_{\text{new}}^{\text{T}} \cdot \mathbf{e}_{\text{new}} \quad . \tag{2.28}$$

As in the case of PCA, the Q_X statistics is assumed to be computed with reconstruction errors from a pre-processed input observation $\mathbf{x}_{\text{new}}$.

The significance of such statistics can be assessed comparing the values of T_X^2 and Q_X to their confidence limits. These can be derived with the same approaches mentioned for PCA in Section 2.1.3. In particular, the F distribution approach and the χ^2 distribution with matching moments can be used for T_X^2, while the Jackson-Musholkad approach and the χ^2 distribution with matching moments can be used for Q_X. More sophisticated approaches free from distributional assumptions exist also for PLS. See Section 7.2.2 for details.

Note that statistics equivalent to T_X^2 and Q_X could be defined for the output variables. However, only statistics related to the input variables are available if the PLS model is applied to a new observation $\mathbf{x}_{\text{new}}$ with the aim to estimate an unknown $\mathbf{y}_{\text{new}}$. Finally, to T_X^2 and Q_X are the basis of quality-relevant monitoring by PLS (Kourti et al., 1995, 1996; Nomikos et al., 1994, 1995a; Qin, 2003; Wise et al., 1996), as discussed in Section 1.2.3. Further details are given in Section 7.2.1.

2.3 Canonical correlation analysis (CCA)

CCA (Hardoon et al., 2004; Hotelling, 1936; Uurtio et al., 2018) is a multivariate statistical method to explore the relationship between two set of variables. CCA is also referred to as canonical variate analysis (CVA), especially in the systems identification literature (Larimore, 1983, 1990). In this Thesis, the name CCA is used to identify the model described in Section 2.3.1, while the name CVA refers to the dynamic extension of CCA, which will be introduced in Section 2.3.4. Furthermore, the literature on CCA and CVA generally discusses the model using the random vector form. In this Thesis, the methods are outlined in their sample form.

2.3.1 Model calibration

Given a matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$ containing N obersvations of V_X input variables and a matrix $\mathbf{Y} \in \mathbb{R}^N \times \mathbb{R}^{V_Y}$ gathering the same number of observations of V_Y output variables, CCA aims at finding A pairs of canonical variables (CV) respecting the following conditions:

- input and output CVs are linear combinations of the input and output variables, respectively;
- one input CV is orthogonal to all other input CVs;
- one output CV is orthogonal to all other output CVs;
- the correlation coefficient between input and output CVs in a pair, called canonical correlation coefficient, is maximized.

CCA provides two matrix projection models:

$$\mathbf{C} = \mathbf{X} \cdot \mathbf{J} \quad , \tag{2.29}$$

$$\mathbf{D} = \mathbf{Y} \cdot \mathbf{L} \quad , \tag{2.30}$$

where:

- $\mathbf{C} \in \mathbb{R}^N \times \mathbb{R}^A$ is the input score matrix, the columns of which represent the input CVs and rows are the projections of the observations in $\mathbf{X}$ onto the space of input CVs;
- $\mathbf{J} \in \mathbb{R}^{V_X} \times \mathbb{R}^A$ is the matrix of input canonical weights, which contains coefficients to formulate the input CVs as linear combinations of input variables;
- $\mathbf{D} \in \mathbb{R}^N \times \mathbb{R}^A$ is the output score matrix, the columns of which represent the output CVs and rows are the projections of the observations in $\mathbf{Y}$ onto the space of output CVs;
- $\mathbf{L} \in \mathbb{R}^{V_Y} \times \mathbb{R}^A$ is the matrix of output canonical weights, which contains coefficients to formulate the output CVs as linear combination of output variables.

The score matrices are orthonormal and allow to compute the canonical correlation coefficients:

$$\mathbf{\Gamma} = \mathbf{C}^{\mathrm{T}} \cdot \mathbf{D} \quad , \tag{2.31}$$

where $\mathbf{\Gamma} \in \mathbb{R}^A \times \mathbb{R}^A$ is a diagonal matrix containing the canonical correlation coefficients on the main diagonal in decreasing order. The rationale of the CVA model is graphically represented in Figure 2.4.

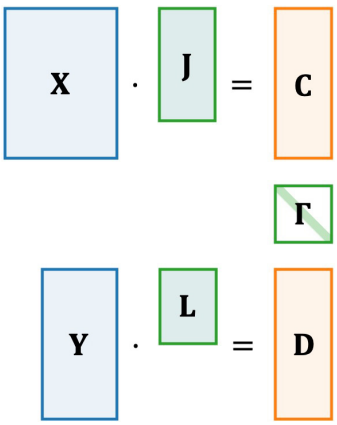


Figure 2.4. *Schematic representation of the CVA model. The matrix projection models in (2.29) and (2.30) are represented by the horizontal equations. The relationship between the CVs in (2.31) is represented vertically.*

Similarly to PLS, CCA is a dimensionality reduction technique as it extracts only A pairs of CVs. However, CCA defines residual CVs as well. Said $R = \min\{V_X, V_Y\}$, a residual model is provided in the form of the matrix projections:

$$\mathbf{C}_{\mathrm{r}} = \mathbf{X} \cdot \mathbf{J}_{\mathrm{r}} \quad , \tag{2.32}$$

$$\mathbf{D}_{\mathrm{r}} = \mathbf{Y} \cdot \mathbf{L}_{\mathrm{r}} \quad , \tag{2.33}$$

where the meanings of the symbols are the same as the ones in equations (2.29) and (2.30), but dimensions of the matrices are now $\mathbf{C}_r \in \mathbb{R}^N \times \mathbb{R}^{R-A}$, $\mathbf{J}_{\mathrm{r}} \in \mathbb{R}^{V_X} \times \mathbb{R}^{R-A}$, $\mathbf{D}_{\mathrm{r}} \in \mathbb{R}^N \times \mathbb{R}^{R-A}$, and $\mathbf{L}_{\mathrm{r}} \in \mathbb{R}^{V_Y} \times \mathbb{R}^{R-A}$. Residual canonical correlation coefficients are defined as:

$$\mathbf{\Gamma}_{\mathrm{r}} = \mathbf{C}_{\mathrm{r}}^{\mathrm{T}} \cdot \mathbf{D}_{\mathrm{r}} \quad , \tag{2.34}$$

where $\mathbf{\Gamma}_{\mathrm{r}} \in \mathbb{R}^{R-A} \times \mathbb{R}^{R-A}$ is a diagonal matrix with the same characteristics of $\mathbf{\Gamma}$. Note that $A < R$ must hold.

The principle of CCA resembles the one of PCA. In CCA, CVs are computed sequentially as to maximize the cross-covariance between the input and output CVs, which are furthermore orthogonal to any other CV in the relevant sequence; PCA relies on a similar rationale, maximizing the covariance modeled by orthogonal PCs on a single data matrix. In fact, also CCA has a strict relationship with the SVD introduced in (2.2). Matrices $\mathbf{X}$ and $\mathbf{Y}$ are assumed to be autoscaled in the following derivation. The sample covariance matrices of $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{S}_X \in \mathbb{R}^{V_X} \times \mathbb{R}^{V_X}$ and $\mathbf{S}_Y \in \mathbb{R}^{V_Y} \times \mathbb{R}^{V_Y}$, respectively, are defined as:

$$\mathbf{S}_X = \frac{1}{\mathrm{N}-1}\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X} \quad , \tag{2.35}$$

$$\mathbf{S}_Y = \frac{1}{\mathrm{N}-1}\mathbf{Y}^{\mathrm{T}} \cdot \mathbf{Y} \quad , \tag{2.36}$$

while the sample cross-covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{S}_{X,Y} \in \mathbb{R}^{V_X} \times \mathbb{R}^{V_Y}$, is given by:

$$\mathbf{S}_{X,Y} = \frac{1}{\mathrm{N}-1}\mathbf{X}^{\mathrm{T}} \cdot \mathbf{Y} \quad . \tag{2.37}$$

Recalling that $R = \min\{V_X, V_Y\}$, the canonical weights can be obtained by first computing the SVD:

$$\mathbf{S}_X^{-\frac{1}{2}} \cdot \mathbf{S}_{X,Y} \cdot \mathbf{S}_Y^{-\frac{1}{2}} = \mathbf{N} \cdot \mathbf{\Sigma} \cdot \mathbf{O}^{\mathrm{T}} = [\mathbf{N}_1 \quad \mathbf{N}_2] \cdot \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \cdot [\mathbf{O}_1 \quad \mathbf{O}_2]^{\mathrm{T}} \quad , \tag{2.38}$$

where, once A (the number of CVs to be retained) is set, the dimension of the matrices involved in the decomposition are similar to the ones elucidated introducing the SVD in Section 2.1.1; matrices $\mathbf{S}_X^{-1/2}$ and $\mathbf{S}_Y^{-1/2}$ can be computed through the Cholesky decomposition (Chapra et al., 2015) of the relevant covariance matrices. Then, the parameters of the CVA model can be obtained as:

$$\mathbf{J} = \mathbf{S}_X^{-\frac{1}{2}} \cdot \mathbf{N}_1 \quad , \tag{2.39}$$

$$\mathbf{L} = \mathbf{S}_Y^{-\frac{1}{2}} \cdot \mathbf{O}_1 \quad , \tag{2.40}$$

$$\mathbf{J}_{\mathrm{r}} = \mathbf{S}_X^{-\frac{1}{2}} \cdot \mathbf{N}_2 \quad , \tag{2.41}$$

$$\mathbf{L}_{\mathrm{r}} = \mathbf{S}_Y^{-\frac{1}{2}} \cdot \mathbf{O}_2 \quad . \tag{2.42}$$

Finally, the score matrices of the residual and main models can be obtained by (2.29), (2.30), (2.32), and (2.33), which in turn allow to obtain the canonical correlation matrices by (2.31) and (2.34). Note that the description given above assumes that $\mathrm{rank}(\mathbf{X}) = V_X$ and $\mathrm{rank}(\mathbf{Y}) = V_Y$. However, the procedure can be easily generalized to the case where the data matrices are not full rank (Russell et al., 2000), for example defining R as $R = \min\{R_X, R_Y\}$, where $R_X = \mathrm{rank}(\mathbf{X})$ and $R_Y = \mathrm{rank}(\mathbf{Y})$.

CCA is strictly connected to PLS as well. Both models are related to SVD: CCA is based on the decomposition of a weighted cross-covariance matrix and aims at maximizing the correlation coefficients between pairs of CVs; on the other hand, PLS performs a sequence of decompositions of the unweighted cross-covariance matrix (working on the full cross-covariance matrix in the first iteration and on residuals from the previous decomposition in the subsequent iterations) and aims at maximizing both the modeled cross-covariance and the variances captured by the data models of the data matrices (Sharper et al., 1994). Furthermore,

PLS can be seen as a penalized version of CCA (Frank et al., 1993): the penalty factors are introduced in the SVD and are basically provided by PCA models of the input and output spaces (Barker et al., 2003).

2.3.2 Model application

An important difference between CCA and PLS is that the latter provides an explicit regression model between the input variables and the output variables. CCA does not provide such a model directly, as it is mostly meant for interpretation and dimensionality reduction. Prediction is possible nonetheless, as outlined in this Section.

Given a new input observation $\mathbf{x}_{\text{new}} \in \mathbb{R}^{V_X}$ (the components of which are assumed to be scaled with the means and standard deviations of the columns of the input matrix **X**), the first step is to project $\mathbf{x}_{\text{new}}$ onto the space of input CVs:

$$\mathbf{c}_{\text{new}}^{\text{T}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{J} \quad . \tag{2.43}$$

The canonical correlation coefficients can then be leveraged to obtain an approximation of the unknown output scores, $\hat{\mathbf{d}}_{\text{new}}^{\text{T}}$:

$$\hat{\mathbf{d}}_{\text{new}}^{\text{T}} = (\mathbf{c}_{\text{new}}^{\text{T}} \cdot \mathbf{c}_{\text{new}})^{-1} \cdot \mathbf{c}_{\text{new}}^{\text{T}} \cdot \mathbf{\Gamma} \quad . \tag{2.44}$$

Finally, the approximated output score can be projected back to the space of output variables solving the least-squares problem:

$$\hat{\mathbf{y}}_{\text{new}}^{\text{T}} = \hat{\mathbf{d}}_{\text{new}}^{\text{T}} \cdot \mathbf{L}^{\text{T}} \cdot (\mathbf{L} \cdot \mathbf{L}^{\text{T}}) \quad , \tag{2.45}$$

where $\hat{\mathbf{y}}_{\text{new}}$ approximates the true, unknown output observation $\mathbf{y}_{\text{new}} \in \mathbb{R}^{V_Y}$ (assumed to be scaled in the same way as matrix **Y**).

2.3.3 Prediction diagnostics

The reliability in prediction of the CCA model can be assessed similarly to PCA, as outlined in Section 2.1.3. A first, elementary index is the projection residual of the input observation $\mathbf{x}_{\text{new}}$ (Russell et al., 2000), defined as:

$$\mathbf{r}_{\text{new}} = \left(\mathbf{I}_{V_X} - \mathbf{J} \cdot \mathbf{J}^{\text{T}}\right) \cdot \mathbf{x}_{\text{new}} \quad . \tag{2.46}$$

Furthermore, three diagnostic statistics can be defined for CCA:

- the T_X^2 statistic (Negiz et al., 1997) measures the squared distance of the projection of a new observation from the center of the input CV space:

$$T_X^2 = \mathbf{c}_{\text{new}}^{\text{T}} \cdot \mathbf{c}_{\text{new}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{J} \cdot \mathbf{J}^{\text{T}} \cdot \mathbf{x}_{\text{new}} \quad ; \tag{2.47}$$

- the Q_X statistic (Russell et al., 2000), measures the squared orthogonal distance between a new observation and the space of CVs:

$$Q_X = \mathbf{r}_{\text{new}}^{\text{T}} \cdot \mathbf{r}_{\text{new}} \quad ; \tag{2.48}$$

- the $T_{X,\text{r}}^2$ statistic (Russell et al., 2000), measures the squared distance of the projection of a new observation from the center of the input CV space in the residual model:

$$T_{X,\text{r}}^2 = \mathbf{c}_{\text{r}_{\text{new}}}^{\text{T}} \cdot \mathbf{c}_{\text{r}_{\text{new}}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{J}_{\text{r}} \cdot \mathbf{J}_{\text{r}}^{\text{T}} \cdot \mathbf{x}_{\text{new}} \quad , \tag{2.49}$$

where $\mathbf{c}_{\mathrm{r_{new}}}^{\mathrm{T}}$ is the projection of $\mathbf{x}_{\mathrm{new}}$ based on the residual model:

$$\mathbf{c}_{\mathrm{r_{new}}}^{\mathrm{T}} = \mathbf{x}_{\mathrm{new}}^{\mathrm{T}} \cdot \mathbf{J}_{\mathrm{r}} \quad . \tag{2.50}$$

The significance of such statistics can be assessed comparing their values to the relevant confidence limits, which can be obtained with the approaches mentioned for PCA in Section 2.1.3. Detailed descriptions can be found in the literature (Martin et al., 1996; Reis et al., 2021a; Russell et al., 2000, 2000; Thissen et al., 2001; Tracy et al., 1992).

2.3.4 Dynamic extension: canonical variate analysis (CVA)

As stated in the introduction to Section 2.3, CCA is generally referred to as CVA in the systems identification literature (Larimore, 1983, 1990). Specifically, CVA is tacitly defined therein as a dynamic generalization of CCA. In fact, CVA found numerous applications to fault detection in dynamic processes (Chiang et al., 2001; Negiz et al., 1997; Russell et al., 2000). Note that dynamic generalizations of PCA and PLS exist as well and will be introduced in Section 7.2.3 in the context of fault detection.

In general, PCA, PLS, and CCA are calibrated starting from covariance and cross-covariance matrices. These entities account for the correlation among variables but neglect the potential correlation among observations, a characteristic of data from dynamic processes (Bergmeir et al., 2012). However, the correlation among observations can be described by the autocorrelation coefficients (Box et al., 2016) and by the cross-correlation coefficients (Brockwell et al., 2016). These coefficients characterize the dynamic behavior of the data; the information they provide can be included in latent-variable models augmenting the data matrices by means of lagged measurements prior to modeling (Ku et al., 1995).

In CVA, matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$ is assumed to contain N obersvations of V_X control inputs of a dynamic system/process; the corresponding observations of the V_Y outputs are assumed to be gathered in matrix $\mathbf{Y} \in \mathbb{R}^N \times \mathbb{R}^{V_Y}$. Such matrices are used to produce the so-called past and future matrices.

The past matrix, $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{N-L-H} \times \mathbb{R}^{(V_X+V_Y)L}$, contains past trajectories of the control inputs and of the outputs over a horizon of extension equal to L observations and is defined as:

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \mathbf{y}_L^{\mathrm{T}} & \mathbf{y}_{L-1}^{\mathrm{T}} & \cdots & \mathbf{y}_1^{\mathrm{T}} & \mathbf{x}_L^{\mathrm{T}} & \mathbf{x}_{L-1}^{\mathrm{T}} & \cdots & \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{y}_{L+1}^{\mathrm{T}} & \mathbf{y}_L^{\mathrm{T}} & \cdots & \mathbf{y}_2^{\mathrm{T}} & \mathbf{x}_{L+1}^{\mathrm{T}} & \mathbf{x}_L^{\mathrm{T}} & \cdots & \mathbf{x}_2^{\mathrm{T}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{N-H-1}^{\mathrm{T}} & \mathbf{y}_{N-H-2}^{\mathrm{T}} & \cdots & \mathbf{y}_{N-H-L}^{\mathrm{T}} & \mathbf{x}_{N-H-1}^{\mathrm{T}} & \mathbf{x}_{N-H-2}^{\mathrm{T}} & \cdots & \mathbf{x}_{N-H-L}^{\mathrm{T}} \end{bmatrix} , \tag{2.51}$$

where $\mathbf{x}_n \in \mathbb{R}^{V_X}$ and $\mathbf{y}_n \in \mathbb{R}^{V_Y}$ represent observations (rows) in the input and output matrices, respectively. The future matrix, $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{N-L-H} \times \mathbb{R}^{V_Y(1+H)}$, contains the future trajectories of the outputs over a horizon of extension equal to H observations and is given by:

$$\boldsymbol{\mathcal{F}} = \begin{bmatrix} \mathbf{y}_{L+1}^{\mathrm{T}} & \mathbf{y}_{L+2}^{\mathrm{T}} & \cdots & \mathbf{y}_{L+H+1}^{\mathrm{T}} \\ \mathbf{y}_{L+2}^{\mathrm{T}} & \mathbf{y}_{L+3}^{\mathrm{T}} & \cdots & \mathbf{y}_{L+H+2}^{\mathrm{T}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{N-H}^{\mathrm{T}} & \mathbf{y}_{N-H+1}^{\mathrm{T}} & \cdots & \mathbf{y}_N^{\mathrm{T}} \end{bmatrix} . \tag{2.52}$$

Note that the first block of V_Y columns in the future matrix is regarded as the "present time", and H "lagged" measurements are added to such block to obtain a total of $H + 1$ blocks of columns in $\boldsymbol{\mathcal{F}}$. The extents of the past and future horizons, L and H, determine the order of the autocorrelation and cross-correlations considered by the CVA model. Note that, for $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{F}}$ to be valid, the following conditions should be verified: $L \geq 1$ and $H \geq 0$. In particular, note that $H = 0$ yields a future matrix containing only the observation at the present time.

Once the past and future matrices are formulated, the CVA model can be calibrated applying the workflow described in Section 2.3.1 for CCA using $\boldsymbol{\mathcal{P}}$ as input matrix and $\boldsymbol{\mathcal{F}}$ as output matrix (in place of **X** and **Y**, respectively). In CVA, the input canonical variables, computed as:

$$\mathbf{M} = \boldsymbol{\mathcal{P}} \cdot \mathbf{J} \quad , \tag{2.53}$$

are named canonical states or process memory. In fact, canonical states are derived accounting for both correlation and autocorrelation in the data, therefore they recount the dynamic evolution of the process (Larimore, 1990). For the same reason, A is known as state order or memory order in CVA.

The calibration of the CVA model requires to set three hyperparameters: the memory order, and the extents of the past and future horizons. Hyperparameters are generally tuned by means of information criteria, such as the Akaike information criterion (Akaike, 1973). Furthermore, as the canonical states capture information on process dynamics, the matrices **X**, **Y**, and **M** can be leveraged to estimate a state-space model of the process. Readers are referred to the literature on CVA for details (Chiang et al., 2001; Larimore, 1990; Russell et al., 2000).

The equations outlined in this Section are sufficient to employ CVA for fault detection (see also Section 7.2.1). Given a new past vector recorded at time k and arranged with the same structure as (2.51):

$$\boldsymbol{\mathcal{p}}_{\text{new}} = [\mathbf{y}_k^{\text{T}} \quad \mathbf{y}_{k-1}^{\text{T}} \quad \cdots \quad \mathbf{y}_{k-L}^{\text{T}} \quad \mathbf{x}_k^{\text{T}} \quad \mathbf{x}_{k-1}^{\text{T}} \quad \cdots \quad \mathbf{x}_{k-L}^{\text{T}}]_{\text{new}}^{\text{T}} \quad , \tag{2.54}$$

the CVA model can be applied to compute the new states at time k:

$$\mathbf{m}_{\text{new}}^{\text{T}} = \boldsymbol{\mathcal{p}}_{\text{new}}^{\text{T}} \cdot \mathbf{J} \quad . \tag{2.55}$$

The diagnostic statistics introduced in Section 2.3.3 hold true also for CVA. However, their interpretation is slightly different (Chiang et al., 2001):

- the T_X^2 statistic describes variations inside the state-space:
$$T_X^2 = \mathbf{m}_{\text{new}}^{\text{T}} \cdot \mathbf{m}_{\text{new}} = \boldsymbol{\mathcal{p}}_{\text{new}}^{\text{T}} \cdot \mathbf{J} \cdot \mathbf{J}^{\text{T}} \cdot \boldsymbol{\mathcal{p}}_{\text{new}} \quad ; \tag{2.56}$$
- the Q_X statistic describes variations in the residual space:
$$Q_X = \mathbf{r}_{\text{new}}^{\text{T}} \cdot \mathbf{r}_{\text{new}} \quad , \tag{2.57}$$
where the residual $\mathbf{r}_{\text{new}}$ is computed as:
$$\mathbf{r}_{\text{new}} = \left(\mathbf{I}_{(V_X+V_Y)L} - \mathbf{J} \cdot \mathbf{J}^{\text{T}}\right) \cdot \boldsymbol{\mathcal{p}}_{\text{new}} \quad ; \tag{2.58}$$
- the $T_{X,\text{r}}^2$ statistic describes variations outside of the state-space:
$$T_{X,\text{r}}^2 = \mathbf{m}_{\text{r}_{\text{new}}}^{\text{T}} \cdot \mathbf{m}_{\text{r}_{\text{new}}} = \boldsymbol{\mathcal{p}}_{\text{new}}^{\text{T}} \cdot \mathbf{J}_{\text{r}} \cdot \mathbf{J}_{\text{r}}^{\text{T}} \cdot \boldsymbol{\mathcal{p}}_{\text{new}} \quad , \tag{2.59}$$
where $\mathbf{m}_{\text{r}_{\text{new}}}^{\text{T}}$ is computed with the residual model:
$$\mathbf{m}_{\text{r}_{\text{new}}}^{\text{T}} = \boldsymbol{\mathcal{p}}_{\text{new}}^{\text{T}} \cdot \mathbf{J}_{\text{r}} \quad . \tag{2.60}$$

Confidence limits can be computed to estimate the significance of the CVA statistics (Chiang et al., 2001; Russell et al., 1998). Additional details on estimation of the confidence limits are discussed in Section 7.2.2.

2.4 Joint-Y partial least-square regression (JYPLS)

Regular PLS is applicable if the available data can be arranged in two matrices **X** and **Y**. However, this is not always possible in some cases of industrial relevance. Considering the example of a biorefinery, the upstream process is carried out by operations traditionally run in batch or fed-batch mode, for example fermentation, while the downstream process adopts operations running in (semi-)continuous regime, as membrane separation and distillation (Bähner et al., 2021). Therefore, an array of parallel units is generally employed in the upstream (for example bioreactors) scheduling the production as to always keep the downstream feed tanks stocked up and guarantee continuous operation of the units therein (Bähner et al., 2021). Even though such array of units is to produce the same product, no guarantee is given on the units themselves to be perfectly identical, or even to be equipped with the same sensors.
The traditional approach to data-driven modeling of parallel units relies on the development of separate models for each unit (Philippe et al., 2013; Shen et al., 2018), possibly aided by methods aimed at assessing differences between single units (Louwerse et al., 1999b). Yet, this method clearly disregards a key information: parallel units should ideally operate in the same way to produce the same product. Besides the additional burden due to the mere presence of multiple models, differences among units could cause models to be significantly different and to yield varying performance. An alternative is the development of a single model for all units under the assumption that all of them produce the same data and operate identically (Tessier et al., 2012), a rather restrictive assumption in real, industrially relevant cases (Louwerse et al., 1999b; Reis et al., 2018; Rendall et al., 2017a). Ideally, a model for an array of parallel units should exploit the relevant information that the product is the same, yet not disregard the potential difference among units. JYPLS (García-Muñoz, 2004; García-Muñoz et al., 2005) is an extension of PLS meant to account for such information.

2.4.1 Model calibration

JYPLS assumes that available data are arranged as two sequences: one for the input variables, $\{\mathbf{X}_1, \dots, \mathbf{X}_P\}$, $\mathbf{X}_p \in \mathbb{R}^{N_p} \times \mathbb{R}^{V_{X_p}}$, and one for the output variables, $\{\mathbf{Y}_1, \dots, \mathbf{Y}_P\}$, $\mathbf{Y}_p \in \mathbb{R}^{N_p} \times \mathbb{R}^{V_Y}$, with $p \in \{1, \dots, P\}$. Each couple of matrices $(\mathbf{X}_p, \mathbf{Y}_p)$ comes from plant p in a set of P plants. The number of observations, N_p, can vary among plants; the same holds for the number of input variables V_{X_p}, which can be in fact different among plants. On the other hand, output variables must be the same for all plants. The latter condition is needed as JYPLS is based on the idea that output variables lay on a common latent space describing the relationship between all the

plants. Therefore, all $\mathbf{Y}_p$ can be jointed (hence the name "joint-Y") in a single matrix $\mathbf{Y}_\mathrm{J} \in \mathbb{R}^{N_\mathrm{J}} \times \mathbb{R}^{V_Y}$, where $N_\mathrm{J} = \sum_{p=1}^{P} N_p$, defined as:

$$\mathbf{Y}_\mathrm{J} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_P \end{bmatrix} \quad . \tag{2.61}$$

On the other hand, input variables can belong to different spaces among different plants, representing within-plant correlations. All data matrices $\mathbf{X}_p$ and $\mathbf{Y}_p$ need to be autoscaled prior to model calibration; furthermore, autoscaled $\mathbf{X}_p$ and $\mathbf{Y}_p$ must be divided by $\sqrt{N_p V_{X_p}}$ and $\sqrt{N_p}$, respectively, to ensure that all plants have the same weights on the model.

The JYPLS model provides P input data models, one for each plant, and only one output data model for the joint space:

$$\mathbf{X}_p = \mathbf{T}_p \cdot \mathbf{P}_p^\mathrm{T} + \mathbf{E}_p \qquad p \in \{1, \dots, P\} \quad , \tag{2.62}$$

$$\mathbf{Y}_\mathrm{J} = \mathbf{T}_\mathrm{J} \cdot \mathbf{Q}_\mathrm{J}^\mathrm{T} + \mathbf{F}_\mathrm{J} \quad , \tag{2.63}$$

where input loadings, $\mathbf{P}_p \in \mathbb{R}^{V_{X_p}} \times \mathbb{R}^A$, and residuals, $\mathbf{E}_p \in \mathbb{R}^{N_p} \times \mathbb{R}^{V_{X_p}}$, can vary across plants; on the other hand, output scores lay on a space of A output LVs common to all plants, therefore the joint score matrix, $\mathbf{T}_\mathrm{J} \in \mathbb{R}^{N_\mathrm{J}} \times \mathbb{R}^A$, can be defined with the same structure of (2.61). The same holds true for the output joint residuals, $\mathbf{F}_\mathrm{J} \in \mathbb{R}^{N_\mathrm{J}} \times \mathbb{R}^{V_Y}$. While these matrices can be split for single plants, the matrix of joint output loadings, $\mathbf{Q}_\mathrm{J} \in \mathbb{R}^{V_Y} \times \mathbb{R}^A$, is the same for all plants.

In JYPLS, weights for each plant, $\mathbf{W}_p \in \mathbb{R}^{V_{X_p}} \times \mathbb{R}^A$, are computed column by column to maximize the modeled cross-covariance between each $\mathbf{X}_p$ and the joint $\mathbf{Y}_\mathrm{J}$. Adjusted weight matrices, $\mathbf{W}_p^* \in \mathbb{R}^{V_{X_p}} \times \mathbb{R}^A$, are then defined for each plant:

$$\mathbf{W}_p^* = \mathbf{W}_p \cdot \left(\mathbf{P}_p^\mathrm{T} \cdot \mathbf{W}_p\right)^{-1} \quad , \tag{2.64}$$

and used to project input observations onto the joint space of LVs:

$$\mathbf{T}_p = \mathbf{X}_p \cdot \mathbf{W}_p^* \quad . \tag{2.65}$$

A schematic representation of the regression model in JYPLS is given in Figure 2.5.

The number of LVs, A, is common to all plants and is usually set to maximize the predictive performance of the JYPLS model on data not used for calibration. Cross-validation with a leave-one-out scheme is the dominant approach (Facco et al., 2014, 2020; Meneghetti et al., 2012; Rudnitskaya et al., 2017). Readers are referred to original sources for details on the model calibration and interpretation procedures (García-Muñoz, 2004; García-Muñoz et al., 2005).

2.4.2 Model application

Once the JYPLS model has been calibrated, a new input observation coming from any of the P plants, $\mathbf{x}_{\mathrm{new}_p} \in \mathbb{R}^{V_{X_p}}$ (assumed to be scaled in the same way as matrix $\mathbf{X}_p$), can be used to predict the corresponding output observation using knowledge from all plants. In fact, $\mathbf{x}_{\mathrm{new}_p}$ is first projected onto the joint space of LVs:

$$\mathbf{t}_\mathrm{new}^\mathrm{T} = \mathbf{x}_{\mathrm{new}_p}^\mathrm{T} \cdot \mathbf{W}_p^* \quad , \tag{2.66}$$

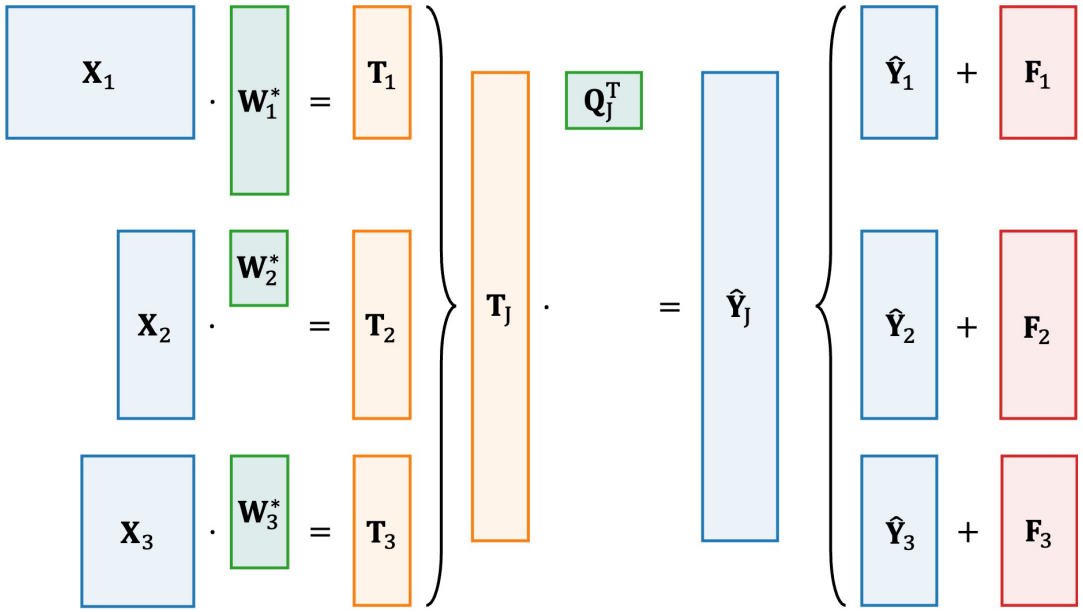


Figure 2.5. *Schematic representation of the JYPLS regression model. The within-plant correlation is captured by the corrected weight matrix for each one of the plants as in (2.65), while the between-plant correlation is accounted for by (2.63): the model of the joint output space. The joint matrices are built as in (2.61).*

and then projected back to the quality space:

$$\hat{\mathbf{y}}_{\text{new}}^{\text{T}} = \mathbf{t}_{\text{new}}^{\text{T}} \cdot \mathbf{Q}_{\text{J}}^{\text{T}} \quad , \tag{2.67}$$

where $\hat{\mathbf{y}}_{\text{new}}$ is an approximation of the true, unknown output observation $\mathbf{y}_{\text{new}} \in \mathbb{R}^{V_Y}$. In principle, the observation $\mathbf{y}_{\text{new}}$ could belong to any of the P plants as it is predicted based on the space of joint output LVs, therefore using information from all the plants. Finally, (2.66) and (2.67) can be jointed in a single equation:

$$\hat{\mathbf{y}}_{\text{new}}^{\text{T}} = \mathbf{x}_{\text{new}_p}^{\text{T}} \cdot \mathbf{W}_p^* \cdot \mathbf{Q}_{\text{J}}^{\text{T}} = \mathbf{x}_{\text{new}_p}^{\text{T}} \cdot \mathbf{B}_p \quad , \tag{2.68}$$

where $\mathbf{B}_p \in \mathbb{R}^{V_{X_p}} \times \mathbb{R}^{V_Y}$ is the matrix of JYPLS regression coefficients to predict an output observation in the joint space starting form input observations from plant p.

2.5 Latent-variable model inversion (LVMI)

While PLS modeling allows to estimate an unknown output observation $\mathbf{y}_{\text{new}}$ given a known input observation $\mathbf{x}_{\text{new}}$, one may also consider the problem of finding $\hat{\mathbf{x}}_{\text{des}}$, the approximation of an unknown input observation $\mathbf{x}_{\text{des}} \in \mathbb{R}^{V_X}$ that should be set to obtain a desired output observation $\mathbf{y}_{\text{des}} \in \mathbb{R}^{V_Y}$ (assumed to be scaled in the same way as the calibration matrix $\mathbf{Y}$). As stated in Section 1.2.4, this problem is particularly relevant to design process conditions given a desired product quality, where $\mathbf{y}_{\text{des}}$ represents a target quality for a product to be manufactured and $\mathbf{x}_{\text{des}}$ are the process conditions that allow to achieve the specified quality (Arce et al., 2021; Jaeckle et al., 1998, 2000; Ruiz et al., 2018; Tomba et al., 2012a, 2013b). The workflow of such operation is illustrated in Figure 2.6 with the example of a PLS model.

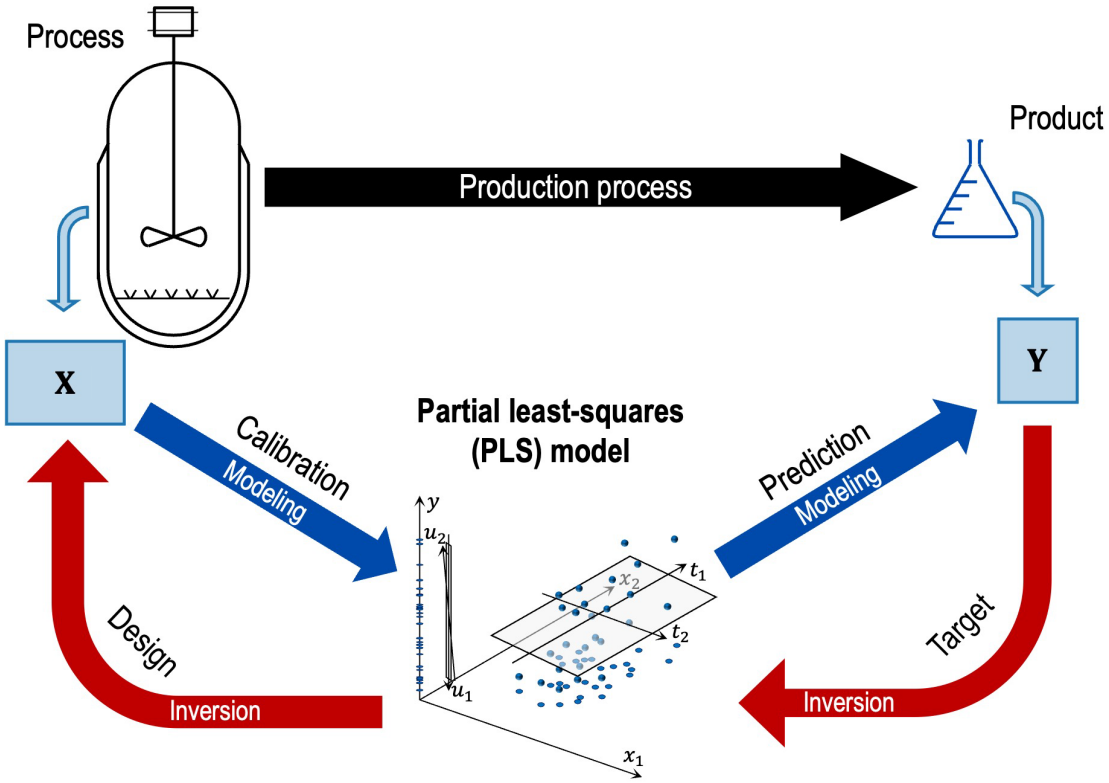


Figure 2.6. *Workflow of LVMI for design of process conditions exemplified on a PLS model. Data characterizing the process and the product are used in the modeling phase for model calibration and use (prediction). In the inversion phase, a target quality is set, and the model is used to design the process conditions that allow to manufacture a product with the desired quality.*

Considering JYPLS, a similar problem could be framed as product transfer between units or plants: one may wish to produce the same product with a given quality on a different process equipment similar, yet not identical, to the one currently being used, or even in an entirely new plant, possibly at a different scale (Dal-Pastro et al., 2017; Facco et al., 2012, 2014; García-Muñoz et al., 2005; Tomba et al., 2014). This kind of problem can be tackled by LVMI.

Focusing on PLS models (without loss of generality), three approaches to LVMI are available:

- algebraic inversion, also referred to as direct inversion (DI), by manipulation of the PLS model equations (Jaeckle et al., 1998, 2000);
- numerical inversion by solution of a nonlinear optimization problem, to minimize the squared difference between the desired output and the model output, possibly subject to both soft and hard constraints on the ranges of input and output variables, as well as on the distances of the solution from the historical process conditions and from the model space (García-Muñoz et al., 2006, 2008; Yacoub et al., 2004);
- numerical inversion by solution of a multi-objective optimization problem aimed at the determination of the Pareto front in the presence of competing targets on multiple output variables in $\mathbf{y}_{\text{des}}$, to analyze the tradeoffs among possible solutions (Arce et al., 2021; Ruiz et al., 2018).

While numerical approaches allow for high flexibility and in-depth analysis of the possible solutions and tradeoffs, this Thesis is concerned with the algebraic approach only due to its mathematical simplicity and computational efficiency (a critical feature in time-sensitive applications, for example real-time control). Therefore, DI (Jaeckle et al., 1998, 2000) is described in detail in this Section, while details on numerical approaches can be found in literature resources (Arce et al., 2021; García-Muñoz et al., 2006, 2008; Ruiz et al., 2018; Tomba et al., 2012a, 2013b; Yacoub et al., 2004). Furthermore, the following discussion regards PLS models, but it can be applied to JYPLS models as well (García-Muñoz et al., 2005).

2.5.1 Direct inversion of PLS models

In the literature regarding LVMI, it is customary to adopt a simplified version of the PLS model. Once matrix $\widetilde{\mathbf{Q}} \in \mathbb{R}^{V_Y} \times \mathbb{R}^{A}$ has been defined as:

$$\widetilde{\mathbf{Q}} = \mathbf{Q} \cdot \mathrm{diag}(\mathbf{b}) \quad , \tag{2.69}$$

PLS models relevant to LVMI are defined as:

$$\widehat{\mathbf{X}} = \mathbf{T} \cdot \mathbf{P}^{\mathrm{T}} \quad , \tag{2.70}$$

$$\widehat{\mathbf{Y}} = \mathbf{T} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}} \quad , \tag{2.71}$$

$$\mathbf{T} = \mathbf{X} \cdot \mathbf{W}^{*} \quad , \tag{2.72}$$

where (2.70) is derived from (2.12) considering only the reconstruction of $\mathbf{X}$ from the PLS data model, therefore setting $\widehat{\mathbf{X}} = \mathbf{X} - \mathbf{E}$, (2.71) derives from the PLS prediction path, jointing (2.18) and (2.19) using matrix $\widetilde{\mathbf{Q}}$ defined as in (2.69), and (2.72) is the same as of (2.15). The prediction path of such simplified PLS model is:

$$\mathbf{t}_{\mathrm{new}}^{\mathrm{T}} = \mathbf{x}_{\mathrm{new}}^{\mathrm{T}} \cdot \mathbf{W}^{*} \quad , \tag{2.73}$$

$$\hat{\mathbf{y}}_{\mathrm{new}}^{\mathrm{T}} = \mathbf{t}_{\mathrm{new}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}} \quad . \tag{2.74}$$

Carrying out LVMI by DI requires some conditions to be set (Jaeckle et al., 1998, 2000): matrix $\mathbf{Y}$ must have more rows than columns, and output variables must be independent. Mathematically, these two conditions imply that matrix $\mathbf{Y}$ is full rank and can be used to obtain a complete basis of its column space as $\mathrm{rank}(\mathbf{Y}) = V_Y$. The discussion on why such an assumption is central to DI will be given in Section 6.2.1. Once the target output $\mathbf{y}_{\mathrm{des}}$ is set (and pre-processed), the first step of DI is to project it onto the space of LVs. Therefore, $\mathbf{y}_{\mathrm{des}}$ is used as $\hat{\mathbf{y}}_{\mathrm{new}}$ in (2.74), which must be inverted to obtain $\mathbf{t}_{\mathrm{des}} \in \mathbb{R}^{A}$. Three cases can arise according to the number of LVs used in the model, and some of them require to use the concept of generalized matrix inverse (Rao et al., 1971).

- If $A < V_Y$, no exact solution exists, but an optimal solution (in the least-squares sense) can be obtained inverting (2.74) by means of the right generalized inverse of $\widetilde{\mathbf{Q}}^{\mathrm{T}}$:

$$\mathbf{t}_{\mathrm{des}}^{\mathrm{T}} = \mathbf{y}_{\mathrm{des}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}^{+\mathrm{R}} = \mathbf{y}_{\mathrm{des}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}} \cdot \left(\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}\right)^{-1} \quad . \tag{2.75}$$

- If $A = V_Y$, an exact solution exists as $\widetilde{\mathbf{Q}}^{\mathrm{T}}$ is square and (2.74) can be inverted directly:

$$\mathbf{t}_{\mathrm{des}}^{\mathrm{T}} = \mathbf{y}_{\mathrm{des}}^{\mathrm{T}} \cdot \left(\widetilde{\mathbf{Q}}^{\mathrm{T}}\right)^{-1} \quad . \tag{2.76}$$

- If $A > V_Y$, an infinite number of solutions to the inversion of (2.74) exist. A particular

solution can be computed using the left generalized inverse of $\widetilde{\mathbf{Q}}^{\mathrm{T}}$:

$$\mathbf{t}_{\mathrm{des,p}}^{\mathrm{T}} = \mathbf{y}_{\mathrm{des}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}^{+\mathrm{L}} = \mathbf{y}_{\mathrm{des}}^{\mathrm{T}} \cdot \left(\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}}\right)^{-1} \cdot \widetilde{\mathbf{Q}} \quad , \tag{2.77}$$

while the complete set of solutions can be found considering the null space (Jaeckle et al., 2000), that is an $(A - V_Y)$-dimensional subspace of the space of LVs which can be represented as:

$$\mathbf{t}_{\mathrm{des,n}}^{\mathrm{T}} = \boldsymbol{\lambda} \cdot \mathbf{G}^{\mathrm{T}} \quad , \tag{2.78}$$

where $\mathbf{G} \in \mathbb{R}^{A} \times \mathbb{R}^{A-V_Y}$ is a matrix containing the $(A - V_Y)$ left singular vectors of $\widetilde{\mathbf{Q}}^{\mathrm{T}}$ as columns, and $\boldsymbol{\lambda} \in \mathbb{R}^{A-V_Y}$ is a vector of arbitrary real numbers. Therefore, the complete set of solutions to the inversion of (2.74) in the case $A > V_Y$ is:

$$\mathbf{t}_{\mathrm{des}}^{\mathrm{T}} = \mathbf{t}_{\mathrm{des,p}}^{\mathrm{T}} + \mathbf{t}_{\mathrm{des,n}}^{\mathrm{T}} = \mathbf{y}_{\mathrm{des}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}^{+\mathrm{L}} + \boldsymbol{\lambda} \cdot \mathbf{G}^{\mathrm{T}} \quad . \tag{2.79}$$

Regardless of the relationship between A and V_Y, the second step of DI is to project $\mathbf{t}_{\mathrm{des}}$ back to the space of input variables by means of (2.70) to obtain $\hat{\mathbf{x}}_{\mathrm{des}}$:

$$\hat{\mathbf{x}}_{\mathrm{des}}^{\mathrm{T}} = \mathbf{t}_{\mathrm{des}}^{\mathrm{T}} \cdot \mathbf{P}^{\mathrm{T}} \quad , \tag{2.80}$$

where $\hat{\mathbf{x}}_{\mathrm{des}}^{\mathrm{T}}$ is clearly a unique solution if $A \leq V_Y$, while it is an $(A - V_Y)$-dimensional subspace (infinite set of solutions) of the space of input variables if $A > V_Y$.

2.5.2 Null space uncertainty

The concept of null space is particularly important when LVMI is used to design new process conditions to achieve a given target quality. In fact, any $\hat{\mathbf{x}}_{\mathrm{des}}$ (or $\mathbf{t}_{\mathrm{des}}$) falling on the null space should yield the same product quality, according to the model (Jaeckle et al., 2000), a property of the null space that has been proven experimentally (Tomba et al., 2014). In this context, the null space represents a degree of freedom to tune the designed process conditions in order to satisfy other objectives, for example the minimization of energy cost, while still obtaining the desired product quality (Jaeckle et al., 2000). The concept of null space has also been observed to be closely related to the concept of design space of a pharmaceutical process (Bano et al., 2018a; Tomba et al., 2012a), which is defined as "the multidimensional combination and interaction of input variables (e.g., material attributes) and process parameters that have been demonstrated to provide assurance of quality" (ICH, 2009). The estimation of the null space uncertainty is of paramount importance when LVMI is used to determine the design space.

Restricting the discussion to PLS model inversion, a number of methods has been proposed to the estimate the uncertainty of the null space. A data resampling approach based on jackknifing has been proposed first (Tomba et al., 2012a). Analytical approaches are available as well, based on uncertainty on prediction (Facco et al., 2015) and uncertainty on model parameters (Bano et al., 2017). Methods to consider observation leverage (Palací-López et al., 2019) or probabilistic formulations of the null space (Bano et al., 2018a) have been proposed as well. Two simple analytical approaches are considered in this Thesis: the one based on PLS prediction uncertainty proposed by Facco et al. (2015), and its extension to include the effect of observation leverage proposed by Palací-López et al. (2019).

The approach by Facco et al. (2015) assumes that the target quality, $\mathbf{y}_{\mathrm{des}}$, can be treated as a prediction from the PLS model; its corresponding projection onto the space of LVs is $\mathbf{t}_{\mathrm{des,p}}$, obtained from (2.77). Therefore, the leverage of $\mathbf{y}_{\mathrm{des}}$ is computed as per (2.23):

$$h_{\mathbf{y}_{\mathrm{des}}} = \frac{\mathbf{t}_{\mathrm{des,p}}^{\mathrm{T}} \cdot \boldsymbol{\Lambda}_T^{-1} \cdot \mathbf{t}_{\mathrm{des,p}}}{N-1} \quad , \tag{2.81}$$

then used to estimate the standard deviation of $\mathbf{y}_{\mathrm{des}}$ from (2.22):

$$\mathbf{s}_{\mathbf{y}_{\mathrm{des}}} = \mathbf{MSE}\sqrt{1 + \frac{1}{N} + h_{\mathbf{y}_{\mathrm{des}}}} \quad , \tag{2.82}$$

and its confidence interval at a given significance level α by (2.21):

$$\mathrm{CI}(\mathbf{y}_{\mathrm{des}}) = \mathbf{y}_{\mathrm{des}} \pm \mathbf{s}_{\mathbf{y}_{\mathrm{des}}} t|_{\frac{\alpha}{2}} \quad . \tag{2.83}$$

The confidence interval of $\mathbf{y}_{\mathrm{des}}$ is then inverted by means of (2.77) as the estimate the confidence interval of $\mathbf{t}_{\mathrm{des,p}}$:

$$\mathrm{CI}\left(\mathbf{t}_{\mathrm{des,p}}^{\mathrm{T}}\right) = \mathrm{CI}\left(\mathbf{y}_{\mathrm{des}}^{\mathrm{T}}\right) \cdot \widetilde{\mathbf{Q}}^{+\mathrm{L}} \quad , \tag{2.84}$$

which are then simply "propagated linearly" as in (2.79) to estimate the confidence interval $\mathbf{t}_{\mathrm{des}}^{\mathrm{T}}$, thus of the null space:

$$\mathrm{CI}\left(\mathbf{t}_{\mathrm{des}}^{\mathrm{T}}\right) = \mathrm{CI}\left(\mathbf{t}_{\mathrm{des,p}}^{\mathrm{T}}\right) + \mathbf{t}_{\mathrm{des,n}}^{\mathrm{T}} \quad . \tag{2.85}$$

It is easy to understand that the method proposed by Facco et al. (2015) relies on the simple inversion of a "constant prediction uncertainty" at $\mathbf{y}_{\mathrm{des}}$ with constant observation leverage estimated at $\mathbf{t}_{\mathrm{des,p}}$. On the other hand, the method proposed by Palací-López et al. (2019) adopts a more sophisticated and theoretically sound approach, which also considers variable observation leverage along the null space. Consider a generic point[2] along the subspace $\mathbf{t}_{\mathrm{des}}$ defined as in (2.79), denoted as $\mathbf{t}_{\mathrm{des}_l}$. The output "data model" in the simplified PLS formulation is first used to reconstruct $\mathbf{y}_{\mathrm{des}}$ using (2.74):

$$\hat{\mathbf{y}}_{\mathrm{des}_l}^{\mathrm{T}} = \mathbf{t}_{\mathrm{des}_l}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}} \quad , \tag{2.86}$$

which is then used to compute the residual associated to the null space:

$$\mathbf{r}_{\mathrm{des}_l} = \mathbf{y}_{\mathrm{des}} - \hat{\mathbf{y}}_{\mathrm{des}_l} \quad . \tag{2.87}$$

Such residual is projected back to the space of LVs according to (2.77) and used to propagate the inversion uncertainty onto the considered point of the null space, defining the "perturbated" scores as:

$$\tilde{\mathbf{t}}_{\mathrm{des}_l}^{\mathrm{T}} = \mathbf{t}_{\mathrm{des}_l}^{\mathrm{T}} + \mathbf{r}_{\mathrm{des}_l}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}^{+\mathrm{L}} \quad . \tag{2.88}$$

The perturbated scores account for the error associated with the null space on the reconstruction of $\mathbf{y}_{\mathrm{des}}$, thus can be used to obtain the leverage of $\hat{\mathbf{y}}_{\mathrm{des}_l}$ as:

$$h_{\hat{\mathbf{y}}_{\mathrm{des}_l}} = \frac{\tilde{\mathbf{t}}_{\mathrm{des}_l}^{\mathrm{T}} \cdot \boldsymbol{\Lambda}_T^{-1} \cdot \tilde{\mathbf{t}}_{\mathrm{des}_l}}{N-1} \quad , \tag{2.89}$$

which allows to estimate the standard deviation of $\hat{\mathbf{y}}_{\mathrm{des}_l}$:

$$\mathbf{s}_{\hat{\mathbf{y}}_{\mathrm{des}_l}} = \mathbf{MSE}\sqrt{1 + \frac{1}{N} + h_{\hat{\mathbf{y}}_{\mathrm{des}_l}}} \quad . \tag{2.90}$$

Finally, (2.21) is leveraged to compute the confidence interval of $\hat{\mathbf{y}}_{\mathrm{des}_l}$ at a given significance

[2] Numerically, this operation would be implemented discretizing the null space in a given number of points by means of $\boldsymbol{\lambda}$ in (2.78) and considering the l-th one.

level α:

$$\mathrm{CI}\left(\hat{\mathbf{y}}_{\mathrm{des}_l}\right) = \hat{\mathbf{y}}_{\mathrm{des}_l} \pm \mathbf{s}_{\hat{\mathbf{y}}_{\mathrm{des}_l}} t|_{\frac{\alpha}{2}} \quad , \tag{2.91}$$

and the confidence interval of the considered null space point, $\mathbf{t}_{\mathrm{des}_l}$, is obtained applying (2.77) to invert the quantity $\pm\mathbf{s}_{\hat{\mathbf{y}}_{\mathrm{des}_l}} t|_{\frac{\alpha}{2}}$:

$$\mathrm{CI}\left(\mathbf{t}_{\mathrm{des}_l}^{\mathrm{T}}\right) = \mathbf{t}_{\mathrm{des}_l}^{\mathrm{T}} \pm \mathbf{s}_{\hat{\mathbf{y}}_{\mathrm{des}_l}} t|_{\frac{\alpha}{2}} \cdot \widetilde{\mathbf{Q}}^{+\mathrm{L}} \quad . \tag{2.92}$$

The procedure can be repeated for any generic points in the subspace $\mathbf{t}_{\mathrm{des}}$ to estimate the confidence intervals on the whole subspace.

The two approaches outlined are compared using the "example data" used by Palací-López et al. (2019) and reported in Table 2.1. The data consist of $N = 6$ observations of $V_X = 5$ input varibales and $V_Y = 1$ output variable. A PLS model is calibrated on autoscaled data with $A = 2$ LVs. The target quality for inversion is set as $\mathbf{y}_{\mathrm{des}} = 204.86$ and inverted by (2.77). A one-dimensional null space exists and its confidence interval is computed with both the approaches by Facco et al. (2015) and Palací-López et al. (2019): results are reported in Figure 2.7(a) and Figure 2.7(b), respectively. The approach by Facco et al. (2015) yields constant confidence limits. On the other hand, Palací-López et al. (2019) considers the observation leverage, therefore the confidence limits show a "hourglass" shape. Also note that the amplitudes of the confidence intervals in correspondence of $\mathbf{t}_{\mathrm{des,p}}$ (the center of the hourglass) obtained with the two approaches are equal.

Table 2.1. *Example dataset from Palací-López et al. (2019) to compare approaches for null space uncertainty estimation in LVMI. The data comprise N = 6 observation of V_X = 5 input variables (x_1, x_2, x_3, x_4, and x_5) and V_Y = 1 output variable (y).*

Observation	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{y}$
1	5.43	7.54	125.64	58.51	50.49	61.85
2	5.43	15.97	126.20	258.48	74.44	278.99
3	99.23	7.54	9893.38	59.29	737.15	307.89
4	99.23	15.97	9765.16	254.11	1576.28	436.40
5	52.33	11.76	2787.64	139.21	583.76	266.08
6	52.33	11.76	2849.95	135.67	630.73	260.52

2.6 Analysis of batch data

All the models mentioned in this Section thus far require data matrices for calibration. Most process data can be arranged in such a way. Considering, for example, continuous processes, it is customary to gather observations in time of many process variables as rows and columns, respectively, of a data matrix. However, data from batch processes pose a different challenge, as a third dimension is usually added: the batch dimension (Nomikos et al., 1994, 1995b). In general, data from a sequence of B batches can be described as a sequence $\{\mathbf{X}_1, \dots, \mathbf{X}_B\}$, where

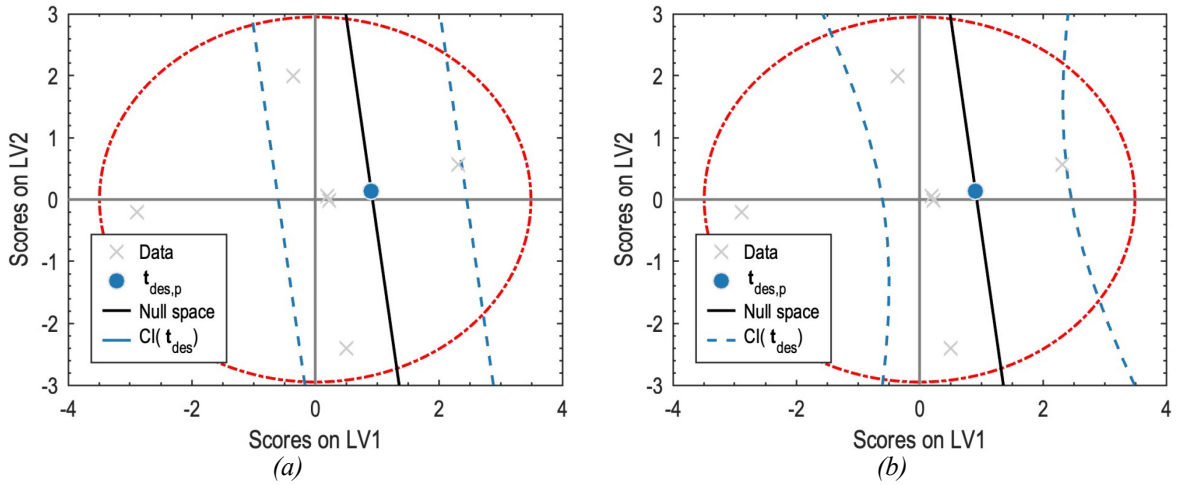


Figure 2.7. *Comparison of approaches proposed by (a) Facco et al. (2015) and by (b) Palací-López et al. (2019) to estimate the uncertainty of the null space in LVMI.*

$\mathbf{X}_b \in \mathbb{R}^{K_b} \times \mathbb{R}^{V_X}$, with $b \in \{1, \dots, B\}$, is a data matrix for batch b containing profiles of the V_X process variables (columns) recorded at K_b times along the batch (rows). Note that the duration of batches can vary. On the other hand, the quality of the product of the batch process is usually measured only at the end of the batch, and can therefore be arranged in a matrix $\mathbf{Y} \in \mathbb{R}^B \times \mathbb{R}^{V_Y}$ collecting the V_Y quality variables for each one of the B batches (Nomikos et al., 1995b).

2.6.1 Multiway methods based on unfolding

If $K_b = K \; \forall \; b \in \{1, \dots, B\}$, meaning that all batches have the same duration, it is customary to arrange the sequence of matrices as a third order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^B \times \mathbb{R}^{V_X} \times \mathbb{R}^K$. Multiway extensions of PCA (Nomikos et al., 1994) and PLS (Nomikos et al., 1995b) allow to analyze such data structure. Alternatively, models equivalent to multiway methods can be calibrated in a simpler way applying conventional PCA and PLS to matrices obtained by batch-wise unfolding (BWU; Wold et al., 1987b). The unfolding is operated "slicing" $\boldsymbol{\mathcal{X}}$ in K matrices $\{\mathbf{X}_1, \dots, \mathbf{X}_K\}$, $\mathbf{X}_k \in \mathbb{R}^B \times \mathbb{R}^{V_X}$, with $k \in \{1, \dots, K\}$, being a matrix containing the records of all V_X variables for all B batches at sampling time k. Matrices in the sequence are then concatenated horizontally to obtain the BWU matrix $\mathbf{X}_{\text{BWU}} \in \mathbb{R}^B \times \mathbb{R}^{KV_X}$, defined as:

$$\mathbf{X}_{\text{BWU}} = [\mathbf{X}_1 \quad \dots \quad \mathbf{X}_K] \quad . \tag{2.93}$$

Such a matrix can be analyzed directly by PCA, or in conjunction with $\mathbf{Y}$ by PLS. However, for a successful application of PCA and PLS to the BWU matrix (in fact, also of their multiway extensions), batches need to be synchronized. This means that all batches have the same duration and key events in the batches happen at the same time. However, the durations of the batches in a sequence are not always the same and can in fact widely vary. Even for batches with equal duration, key process events might be misaligned (González Martínez et al., 2014b). Considering the general case of uneven batch duration, where data are given as $\{\mathbf{X}_1, \dots, \mathbf{X}_B\}$, with $\mathbf{X}_b \in \mathbb{R}^{K_b} \times \mathbb{R}^{V_X}$, one can still obtain a two-dimensional matrix by applying the so-called

variable-wise unfolding (VWU; Wise et al., 1999; Wold et al., 1998). This procedure yields a matrix $\mathbf{X}_{\mathrm{VWU}} \in \mathbb{R}^{\sum_{b=1}^{B} K_b} \times \mathbb{R}^{V_X}$ obtained stacking vertically all matrices in the sequence:

$$\mathbf{X}_{\mathrm{VWU}} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_B \end{bmatrix} \quad . \tag{2.94}$$

PCA can still be applied to $\mathbf{X}_{\mathrm{VWU}}$ to explore the behavior of process variables in time over the sequence of all the B batches. However, PLS is harder to apply in this case, as quality variables need to be measured at the same frequency as process variables, which is not always possible. Finally, BWU and VWU can be seen as extreme cases of the augmentation of matrix $\mathbf{X}_k$ with lagged measurements. In fact, they can be generalized in the so-called batch dynamics unfolding (Chen et al., 2002). This unfolding method shares some properties with the lagged-variables augmentation discussed for CVA in Section 2.3.4. Furthermore, it is the basis for the extension of PCA and PLS to dynamic data, which are discussed in Section 7.2.3.

The application of PCA and PLS to unfolded matrices has been widely studied, and properties of models obtained on different unfolding methods exhaustively discussed in the literature. Interested readers are referred to notable literature resources for more information (Bro et al., 2003; Camacho et al., 2008a, 2009; Gurden et al., 2001; Westerhuis et al., 1999).

If batch data lacks synchronization, a proper multiway analysis can still be applied by first synchronizing the data. Many methods to accomplish this task have been proposed in the literature. Some examples are: truncation of trajectories (Rothwell et al., 1998); extension of trajectories with mean values (Lakshminarayanan et al., 1996); indicator variables (García-Muñoz et al., 2003; Nomikos et al., 1994); dynamic time warping (Kassidas et al., 1998); correlation-optimized time warping (Fransson et al., 2006); relaxed-greedy time warping (González-Martínez et al., 2011); multisynchro (González Martínez et al., 2014a). An exhaustive treatment of synchronization approaches is not within the scope of this Thesis. Readers are referred to the cited studies for details on each approach.

2.6.2 Feature-oriented models

While the synchronization-unfolding path is the preferred approach to batch data analytics, complicated synchronization procedures are needed, and complex models are generally obtained (Rendall et al., 2019). Furthermore, synchronization can yield unsatisfactory results when the duration of batches vary widely and/or profiles of variables exhibit remarkable difference in shapes (Klimkiewicz et al., 2016). This is the case, for example, of membrane filtration processes used in the downstream of biorefineries: membrane fouling causes strong variabilities in batch duration and profile shapes (Abels et al., 2013; Klimkiewicz et al., 2016; Maere et al., 2012; Naessens et al., 2017). Feature-oriented modeling (Reis et al., 2022; Rendall et al., 2019; Yoon et al., 2001) offers an elegant way to address the lack-of-synchronization issue, while also emphasizing the phenomena one wants to model by properly defining features.

The fundamental idea on feature-oriented modeling is to employ feature engineering methods (Bakshi et al., 1996; Reis et al., 2022; Stephanopoulos et al., 1997) to obtain meaningful numerical indices from time profiles of process variables. Formally stated, feature synthesis can be interpreted as an operator:

$$\mathcal{F}: \mathbb{R}^{K_b} \times \mathbb{R}^{V} \rightarrow \mathbb{R}^{F} \mid \mathbf{X}_b \mapsto \mathbf{h}_b \quad , \tag{2.95}$$

where $\mathbf{h}_b \in \mathbb{R}^F$ is a vector containing values of F features characterizing batch b. As one vector of features is obtained from each batch, a matrix $\mathbf{H} \in \mathbb{R}^B \times \mathbb{R}^F$ gathering features (columns) for each batch (rows) can be built and modeled directly by PCA or as the input data in PLS. A comparison between the customary synchronization-unfolding approach and the feature-oriented approach to handle batch data is schematically presented in Figure 2.8.

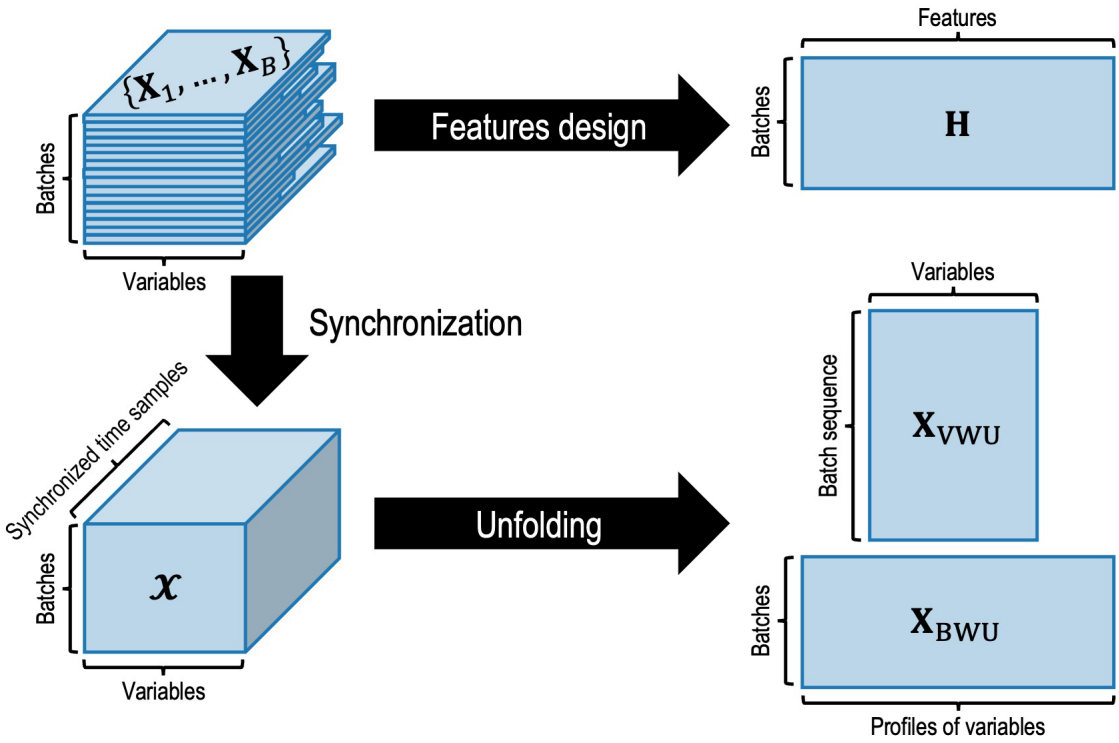


Figure 2.8. *Comparison of approaches for batch data analytics in the case of lack-of-synchronization of batches: synchronization-unfolding path and feature-oriented modeling.*

Several methods have been proposed to synthesize features from profiles of process variables. The first and most natural approach is based on the so-called knowledge-driven features (Wold et al., 2009), also called landmark features (Rendall et al., 2019), where features are defined exploiting process knowledge and simple mathematical operations. Considering the example of a membrane filtration process where the feed pressure is increased along the batch to keep a constant permeate flux counteracting fouling, informative features may be the average and maximum pressure, or the average pressure slope, which is intuitively related to the fouling-rate (Monclús et al., 2011; Naessens et al., 2017). It is easy to understand that a proper definition of knowledge-driven feature could significantly aid model interpretation and process

understanding, even to the price of losing or attenuating information localized on specific time intervals in profiles of variables (Rendall et al., 2019). Knowledge-driven features are the method of interest in this Thesis. However, other methods for feature generation exist.

- Wavelet-based features (Bakshi et al., 1996; Stephanopoulos et al., 1997): features are the coefficients of the multi-scale wavelet decomposition of profiles of variables.
- Statistical pattern analysis (He et al., 2011; Wang et al., 2010): features are defined as statistical moments of profiles of variables.
- Translation-invariant multiscale energy-based features (Rato et al., 2017): features are derived from the wavelet decomposition of profiles of variables.
- Profile-driven features (Rendall et al., 2017a): profiles of variables are matched with and fitted to specific parametric archetypes in a given profile library, and parameters of the fitted archetypes are used as feature.

In this Thesis, feature-oriented methods are intended as the ones mentioned above, where time profiles of process variables are "summarized" into a set of scalar variables. A detailed description of other methods is not within the scope of this Thesis. Readers are referred to the cited references and other relevant literature resources (Reis et al., 2018, 2022; Rendall et al., 2019). Although not relevant for this Thesis, it is also worth noting that features can be extracted for other reasons, such as for batch data synchronization (Andersen et al., 2012) or as implicit nonlinear transformations of scalar variables, for example in the kernel methods paradigm (Cremers et al., 2003; Müller et al., 2001; Pilario et al., 2020; Schölkopf et al., 1999).

2.7 Hybrid models

All the mathematical methods discussed so far are based exclusively on data and do not require any knowledge on the process that generated the data. However, the integration of such knowledge may be beneficial to the modeling exercise and aim. One such way to introduce this knowledge is the feature-oriented paradigm just discussed, specifically data augmentation by feature engineering: process knowledge can be leveraged to synthesize additional informative variables to be added to the dataset by means of first-principles/mechanistic models, then the augmented dataset is used to develop a data-driven model (Destro et al., 2020; Ghosh et al., 2021; Yoon et al., 2001). A different way to combine data-driven and knowledge-driven models is by means of hybrid modeling (Narayanan et al., 2023; Rajulapati et al., 2022; Sansana et al., 2021; Solle et al., 2017; von Stosch et al., 2014; Yang et al., 2020).

In their most common conception, the idea underlying hybrid models is to establish a first-principles modeling framework and to insert data-driven elements into it (Sansana et al., 2021), to be used to describe phenomena that cannot be modeled by means of theoretical tools (Narayanan et al., 2019). This can be achieved combining the "blocks" in multiple ways. In general, two main structures can be defined (Sansana et al., 2021; von Stosch et al., 2014). In

the serial structure, shown in Figure 2.9(a), the knowledge-based model and the data-driven model are connected in series, meaning that the input of one block are the outputs of the other one. In chemical engineering applications, especially concerning soft sensing, the data-driven model is most commonly the first one in the serial structure, and its outputs serve as inputs of the knowledge-driven model. This structure is adopted when there is the complete lack of theoretical knowledge on some of the phenomena to be modeled. In the parallel structure, reported in Figure 2.9(b), both models receive the same inputs, then their outputs are combined by summation, multiplication, or by fancier weighted combinations. This structure is particularly attractive when the knowledge-based model can approximately represent the phenomena of interest, therefore the data-driven element acts as a simple correction.

While the two examples discussed assume that there are a single data-driven element and a single knowledge-driven model, more complex structures can be devised if multiple blocks are used. A relevant one is the so-called combined parallel-serial structure (Teixeira et al., 2005), represented in Figure 2.9(c). This structure is particularly relevant in hybrid modeling of bioreactors. Concentrations of biomass, products, and other species of interest are used as inputs to an approximate kinetic model and to a data-driven model in parallel; outputs are combined by multiplication to obtain the transformation rates of all species considered in the model, which are then fed to a second knowledge-driven block: the material balances of the reactor.

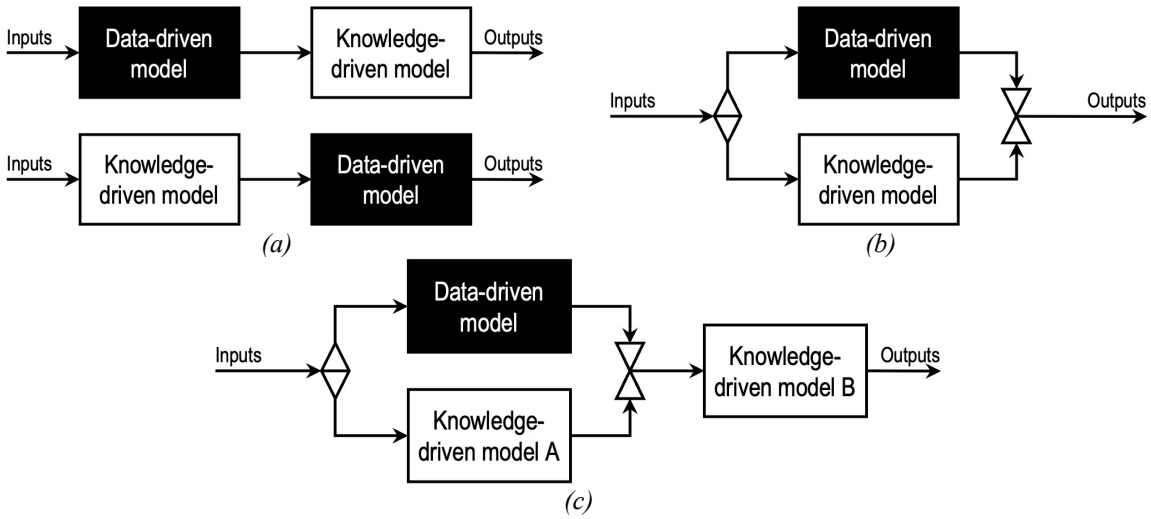


Figure 2.9. *Possible architectures of hybrid models. (a) Serial structure, (b) parallel structure, and (c) parallel-serial structure.*

In (bio-)chemical engineering applications, the knowledge driven part is generally given by material and energy balances, kinetic models, phase equilibrium and transport laws, or sophisticated cell biology models. The data-driven model is generally a high complexity machine learning method, such as artificial neural network or support vector regression. This choice is due to the fact that the data-driven element is usually responsible for the flexibility of the hybrid model, therefore nonlinear methods (possibly providing the universal approximation

property) are appropriate choices (von Stosch et al., 2014). Artificial neural networks are particularly popular (Chen et al., 1995; Oliveira, 2004; Sansana et al., 2021; von Stosch et al., 2014): they were used in the first hybrid model proposed in the literature (Psichogios et al., 1992), and subsequently in many application concerning bioreactors (Marques et al., 2017; Psichogios et al., 1992; Schubert et al., 1994; Teixeira et al., 2005; Vande Wouwer et al., 2004; von Stosch et al., 2016) and membrane separation processes (Chan et al., 2017; Chew et al., 2017; Grisales Díaz et al., 2017; Hwang et al., 2009; Piron et al., 1997). The use of latent-variable models in the hybrid modeling context has been explored by few studies (Carinhas et al., 2011; Destro et al., 2020; Ghosh et al., 2021; Henneke et al., 2005; Lee et al., 2005; Reis et al., 2023; von Stosch et al., 2011).

The most notable advantage of hybrid models over purely data-driven models is increased reliability in general (Narayanan et al., 2019; von Stosch et al., 2014): the data-driven element brings flexibility for adapting to different scenarios, while the knowledge-driven element increases the model robustness. With respect to purely data-driven model, calibration of the data-driven element in a hybrid model often requires less data and lower complexity to achieve a comparable accuracy. Last but not least, hybrid models are more reliable when it comes to respect the underlying physical principles of the process, which is a common issue with data-driven models and is particularly important if extrapolation is required (Raissi et al., 2019).

Chapter 3

Data-driven analysis and improvement of the upstream bioconversion process

A structured application of data-driven methods aimed at understanding and improving the bioconversion step in the upstream process is presented in this Chapter. PCA is used to investigate the differences among units in an array of seven parallel bioreactors. JYPLS regression is leveraged to address a decreasing trend in the quality of the final product, then combined with LVMI to develop guidelines for recovering the process from the quality loss.

3.1 Introduction

The key to sustainability of biorefineries relies in the raw materials: biomass from renewable sources. While many so-called biorefinery platforms exist (Cherubini, 2010; Gavrilescu, 2014; Ubando et al., 2020), bioconversion-based biorefineries are particularly relevant to the process industry due to their potential to produce a wide range of fuels and chemicals (Cuellar et al., 2020; Rosales-Calderon et al., 2019). Fermentation is the most common bioconversion technology in biorefineries (Woodley, 2020). In the typical process, biomass is first prepared for processing: sugar-rich biomass can be fermented directly, while starch-rich biomass must be pre-treated to release the fermentable sugars (Delbecq et al., 2018; Ubando et al., 2020). Sugars are fed to the bioconversion reactor, typically operating in (fed-)batch mode (Bähner et al., 2021). The product mixture processed in the bioreactor is sent to product recovery and purification in the downstream section after batch completion (Cuellar et al., 2020).

Due to the downstream including many continuous operations, such as distillation, it is common to adopt arrays bioreactors operating in parallel in the upstream section to continuously feed the downstream. This strategy is typical for batch processing in general (Louwerse et al., 1999b; Shen et al., 2018) and is also adopted in special downstream units naturally operating in semi-continuous mode due to phenomena such as membrane fouling or resin exhaustion (Zydney, 2016). However, parallel units entail a number of complications, the most prominent ones regarding production scheduling (Bähner et al., 2019). Precise operation and monitoring of each one of the parallel units are of paramount importance for the array itself and for the subsequent units, as scheduling issues in the upstream may be propagated to the downstream (Bähner et al., 2021). Subtler drawbacks regard potential differences among units: while parallel units are

generally meant to operate in the exact same way and to yield the same performance, such conditions are not always met in the industrial practice (Louwerse et al., 1999b; Philippe et al., 2013; Rendall et al., 2017a).

The operation of arrays of parallel units can be supported by process systems engineering and, more in general, mathematical modeling. The potential of these approaches has been highlighted by several studies in the context of biorefineries and bioprocessing (Bähner et al., 2021; Culaba et al., 2022; Velidandi et al., 2023). Data-driven models are of particular interest in industrial environments due to the massive dataset generally produced daily in modern plants (Cuellar et al., 2020). Latent-variable models, such as PCA and PLS, found many successful applications in scenarios involving parallel units as well. Louwerse et al. (1999b) employed PLSDA to understand the difference among two apparently identical batch polymerization reactors, identifying potential causes of unexpected different performances of the two units. A similar problem was tackled by Rendall et al. (2017a): they investigated differences between two industrial dryers in a crystallization process by feature-oriented modeling, achieving similar results. Tessier et al. (2012) adopted multiblock PLS to model an array of thirty-one parallel reactors for aluminum reduction, with the aim to set up a process monitoring system; they developed a single monitoring model for all the units in the array. Philippe et al. (2013) considered an array of four industrial membrane bioreactors for wastewater treatment and developed separate PLS models for each one of them aiming at predicting the dynamic evolution of the membrane resistances to permeation; they noted remarkable, unexpected differences in prediction performance of the models, despite all the units being identical in principle. Shen et al. (2018) proposed to merge the two approaches (a single model for all units or a separate model for each one of the units). They considered a system of three simulated fed-batch reactors in parallel and developed a multi-layer monitoring system composed of: a PCA model for the whole array of reactors for general monitoring; single PLS models for each one of the reactors for quality-relevant monitoring, to account for potential difference not captured by the global PCA model. Their system yielded good performance, but also featured a remarkable complexity.

In light of these examples, it is easy to identify two intuitive approaches to model arrays of parallel units. One can assume that all the units behave in the same way (meaning that they are characterized by the same variables and share the same correlation structure) and develop a single model for all of them; however, such an assumption could not be met in practice and can yield unsatisfactory performance in the presence of differences among units. Alternatively, multiple models can be developed, one for each unit in the array; no assumption on the similarity among units is required, but this approach could become cumbersome nonetheless when the number of parallel units is large. Furthermore, the latter approach effectively neglects the similarities among units and the valuable information that they should be driven by the same fundamental phenomena. This is precious information nonetheless, especially in cases where a

model-based analysis is to be carried out on arrays of reactors manufacturing the same product, for example for soft sensing or data-driven process improvement.

An alternative approach relies on modeling strategies designed to handle arrays of parallel units. JYPLS (García-Muñoz, 2004; García-Muñoz et al., 2005) is one such model. JYPLS was originally proposed to tackle product transfer problems (García-Muñoz et al., 2005), either between units or between entire plants. Many successful applications of JYPLS have been published, including root cause analysis in process development (García-Muñoz et al., 2009), product transfer between units (Tomba et al., 2014) and plants (Chu et al., 2021), transfer of process monitoring models (Facco et al., 2012, 2014), and process scale-up (Dal-Pastro et al., 2017; Facco et al., 2020; Liu et al., 2011b).

However, the potential of JYPLS in scenarios involving arrays of supposedly identical parallel units meant to manufacture the same product has been barely explored (Rudnitskaya et al., 2017). JYPLS is a particularly appealing choice in this case due to its ability to model both phenomena common to all units (between-unit correlation) and phenomena specific of single units (within-unit correlation) (García-Muñoz, 2004; García-Muñoz et al., 2005). In this sense, JYPLS can be used for data fusion (Azcarate et al., 2021), which is particularly relevant in cases where little data are available for each unit: models of single units would be prone to overfitting, while a single model for the whole array could not yield satisfactory performance due to potential differences among units.

In this Chapter, we present a structured analysis of the bioconversion step in the upstream process of the industrial biorefinery considered in this Thesis (Novamont S.p.A., 2016). The operation involves seven bioreactors, which are regarded to operate equivalently despite featuring minor equipment differences. The study is carried out under small data conditions (data on few batches available for each unit). We first carry out a data-driven analysis to spot potential differences among the units using real data of two and a half months of stable operation of the plant. We then investigate a generalized decreasing trend in the end-of-batch concentration of the main product affecting all the bioreactors and captured by a second dataset covering three months of operation. We demonstrate the potential of the selected data-driven method to effectively handle complex industrial processes featuring parallel units when little data are available for each one of them, producing significant process understanding and offering precious guidelines to improve process operation.

However, while we believe in the value of our approach, we cannot provide any experimental evidence of its effectiveness. Significant process/equipment changes took place in the plant set-up shortly after the completion of this study, which conflicted with the implementation of the guidelines we proposed. The changes were motivated by reasons other than the problem we investigated and caused large variations in the operation of the upstream process, including in the reference values of the quality variables. This effectively prevented us from testing the results of our approach on the plant, and from verifying the resolution of the issue of decreasing

end-of-batch quality, which motivated this study in the first place. However, in light of the evidence we present in this Chapter, we are confident that our approach offers a good solution to the problem we investigated.

The reminder of this Chapter is organized as follows. The process and the available datasets are introduced in Section 3.2, wherein the objective of the analysis is stated as well. Section 3.3 outlines the fundamentals of the mathematical methods relevant to this study. Results concerning the assessment of differences among bioreactors are discussed in Section 3.4, while Section 3.5 addresses the decreasing trend of end-of-batch product quality, offering some guidelines for process recovery. Finally, conclusions of this study are drawn in Section 3.6.

3.2 Bioconversion process and data

The bioconversion process is described in this Section. The available dataset and the decreasing trend in the end-of-batch product quality, the troubleshooting of which is the aim of the work described in this Chapter, are illustrated as well.

3.2.1 Bioconversion step

The bioconversion step is at the heart of the biorefinery upstream process and relies on seven bioreactors operating in parallel. In this Chapter, the seven bioreactors will be identified as BRp, $p \in \{1, \dots, 7\}$. A simplified scheme of a bioreactor is reported in Figure 3.1.

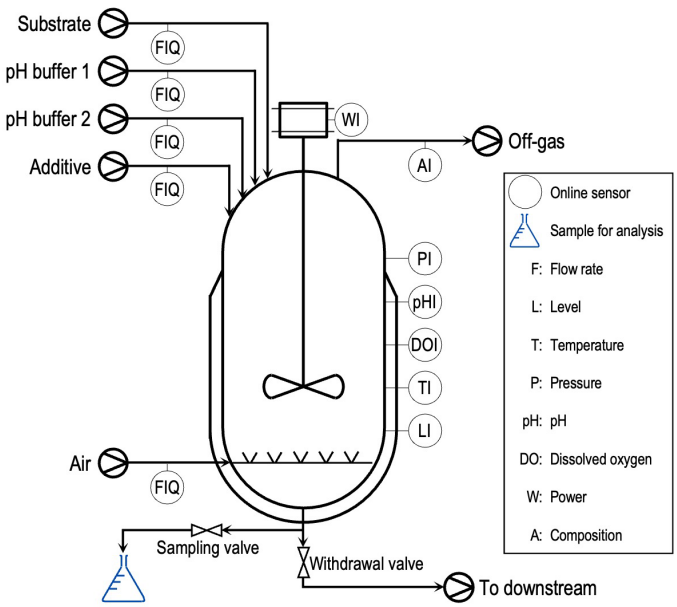


Figure 3.1. *Simplified scheme of a bioreactor in the upstream process.*

The array of bioreactor operates in cycled fed-batch mode (Zydney, 2016) to ensure continuous feed to the downstream process. Batches are recipe-driven and generally have a predefined and consistent duration.

In a typical batch, the bioreactor is first loaded with the culture medium and a given amount of substrate, then it is inoculated with the microorganism. The mixing system is activated, and the microorganism is grown at constant aeration rate (inlet flow rate of the air sparged at the bottom of the bioreactor). When a preset level of DO is achieved, the batch is conducted for a set time at such a constant value. The oxygen uptake rate (OUR; see Section 3.2.3) is used to monitor the biomass growth and the progression of the batch. Substrate and an additive substance[3] having an effect on the gas-to-liquid mass transfer are fed throughout the batch according to a given feeding schedule. The pH is controlled by addition of two different pH buffers (identified as 1 and 2 in Figure 3.1). Buffer 1 is used in the first part of the batch, until a shift in the set-point of the pH is triggered at a preset time. When the pH reaches the new setpoint, buffer 2 is used up to completion of the batch. The bioreactor is then withdrawn, its content sent to a buffer tank before downstream processing. Profiles of process variables related to respiration and pH for a typical batch are reported in Figure 3.2. All variables will be reported as normalized within the $[0, 1]$ interval in this Chapter due to confidentiality reasons.

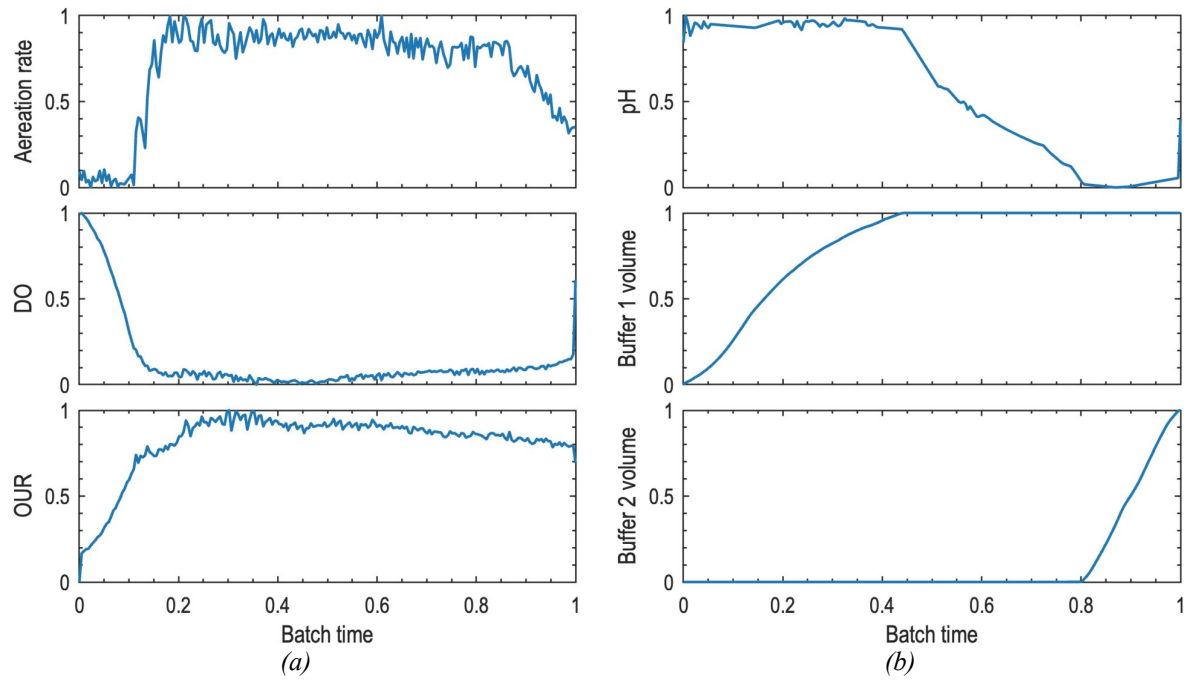


Figure 3.2. *Example of (a) respiration-related variables and (b) pH-related variables in a typical batch of the bioconversion process.*

3.2.2 A decreasing trend in the end-of-batch product concentration

A sampling valve is located on the withdrawal pipe of each bioreactor to allow the plant personnel to collect samples for end-of-batch quality analysis, carried out in the quality control laboratory according to standardized procedures. The two most important quality attributes are the final BDO concentration and a process-specific quality attribute, the nature of which is

[3] The chemical nature of the additive will not be disclosed due to confidentiality reasons.

confidential and will thus be generically denoted as “quality index” throughout this Chapter. While the former quality attribute should be as high as possible, the quality index is of paramount importance for the downstream process and should be as low as possible.
A decreasing trend in the end-of-batch BDO concentration has been detected by plant operators. An increasing trend in the quality index has been detected as well, developing simultaneously to the product concentration trend. Figure 3.3 shows that the trends affect all the bioreactors.

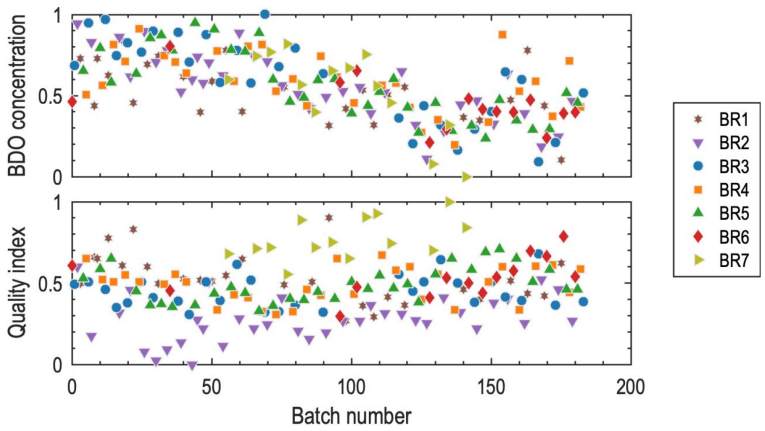


Figure 3.3. *Trend of (a) BDO concentration and (b) quality index across several consecutive batches. Each point represents the value of the relevant quality attribute measured at the end of a batch.*

Figure 3.3 also highlights that the quality variables suffer from high variability. This variability is mostly due to the production process itself and to the fact that multiple bioreactors are used. This is apparent in the quality index, where different units seem to yield products with different values of such variable. The analyses carried out by the plant personnel follow standardized procedures, thus the measurement-induced variability is regarded as a minor contribution to the total one. Despite the high variability, the trends in the quality variables are clearly visible.
Some corrective actions have been implemented by plant operators by trial and error, which allowed to partially recover the BDO concentration (but not to counteract the increase of the quality index), as seen in the last portion of Figure 3.3 (approximately from observation no. 140 onwards). However, the causes of such trends remain unclear. Therefore, we carry out a data-driven investigation to shed some light on the likely causes of the trends. We also aim at developing guidelines for data-driven process improvement, specifically in the form of corrective actions to be implemented in the future to recover the product quality.

3.2.3 Available dataset

Figure 3.1 reports the online sensors installed on bioreactors (all of them feature the same sensors). The main body features sensors for temperature, pressure, level (which allows to infer the volume), DO, and pH. The power required by the mixing system is measured as well. Volume totalizers are installed on the inlet pipes of air, substrate, pH buffers, and additive. An

analyzer is connected to the off-gas outlet and delivers instantaneous measurements of the mole-fractions of oxygen, nitrogen, carbon dioxide, and water in the off-gas.

Besides the measured variables, some additional variables are computed and monitored throughout the batch by virtue of the valuable information they provide. Inlet volume-flow rates are computed from added volumes as:

$$\dot{V}_i(k) = \frac{V_i(k) - V_i(k-1)}{\Delta t} \quad , \tag{3.1}$$

where $\dot{V}_i(k)$ represents the i-th inlet volume-flow rate $[\mathrm{m^3 s^{-1}}]$ at time k, $V_i(k)$ and $V_i(k-1)$ denote the i-th added volume $[\mathrm{m^3}]$ at times k and $k-1$, respectively, and Δt is the sampling interval [s]; i can be air, substrate, pH buffer 1, pH buffer 2, or additive.

The OUR represents the rate $[\mathrm{mol\ s^{-1}}]$ at which microorganisms consume the oxygen provided by the inlet air and is defined as:

$$\mathrm{OUR} = \frac{\dot{V}_{\mathrm{air}}}{\tilde{V}} r_{\mathrm{i}} \left(y_{\mathrm{O_2}}^{\mathrm{in}} - y_{\mathrm{O_2}}^{\mathrm{out}} \right) \quad , \tag{3.2}$$

where $\tilde{V}$ represent the mole-specific volume $[\mathrm{m^3 mol^{-1}}]$ of air, while $y_{\mathrm{O_2}}^{\mathrm{in}}$ and $y_{\mathrm{O_2}}^{\mathrm{out}}$ are the mole-fractions $[-]$ of oxygen in the inlet air and in the off-gas, respectively; furthermore, r_{i} accounts for the water vapor entrained in the off-gas by the gas flowing through the liquid (Mainka et al., 2019) and is defined as:

$$r_{\mathrm{i}} = \frac{1 - y_{\mathrm{O_2}}^{\mathrm{in}} - y_{\mathrm{CO_2}}^{\mathrm{in}}}{1 - y_{\mathrm{O_2}}^{\mathrm{out}} - y_{\mathrm{CO_2}}^{\mathrm{out}} - y_{\mathrm{H_2O}}^{\mathrm{out}}} \quad . \tag{3.3}$$

Finally, the carbon dioxide evolution rate (CER) represents the rate $[\mathrm{mol\ s^{-1}}]$ at which microorganisms release $\mathrm{CO_2}$ as product of their metabolism. The CER is defined as:

$$\mathrm{CER} = \frac{\dot{V}_{\mathrm{air}}}{\tilde{V}} r_{\mathrm{i}} \left(y_{\mathrm{CO_2}}^{\mathrm{out}} - y_{\mathrm{CO_2}}^{\mathrm{in}} \right) \quad , \tag{3.4}$$

where $y_{\mathrm{CO_2}}^{\mathrm{in}}$ and $y_{\mathrm{CO_2}}^{\mathrm{out}}$ are the mole-fractions $[-]$ of carbon dioxide in the inlet air and in the off-gas, respectively. The process variables used is this study are summarized in Table 3.1.

The quality of the final product is characterized at the end of each batch by laboratory analyses. Quality attributes include: concentration of BDO; quality index; biomass concentration expressed as optical density (OD); residual concentration of substrate (denoted as S); concentrations of three byproducts of the bioconversion (identified as Bi, $i \in \{1, \dots, 3\}$); concentrations of seven ionic species in the solution (identified as Ii, $i \in \{1, \dots, 7\}$). In particular, I1 and I2 are ionic species released by pH buffers 1 and 2, respectively, while the remaining species are introduced in the bioreactors with the culture medium.

Two more quality variables are computed for the valuable information they provide on the performance of the process. The BDO productivity $[\mathrm{mol\ m^{-3}\ s^{-1}}]$ is defined as the final BDO concentration divided by the batch time:

$$\mathrm{Pr} = \frac{c_{\mathrm{BDO}}}{t_{\mathrm{tot}}} \quad , \tag{3.5}$$

where c_{BDO} is the final BDO concentration $[\mathrm{mol\ m^{-3}}]$, and t_{tot} is the batch duration [s]. Finally, the yield is defined as:

$$Y = \frac{V M_{\mathrm{BDO}} c_{\mathrm{BDO}}}{m_{\mathrm{S}}^{\mathrm{c}}} \quad , \tag{3.6}$$

hence as the ratio between the total mass [kg] of BDO produced (that is $VM_{\mathrm{BDO}}c_{\mathrm{BDO}}$, V being the volume [m^3] measured at the end of the batch and M_{BDO} the molecular weight [$\mathrm{kg\ mol}^{-1}$] of BDO) and the total mass [kg] of substrate consumed, $m_{\mathrm{S}}^{\mathrm{c}}$. The product quality variables used is this study are listed in Table 3.2.

Process data are provided as time profiles of the 22 process variables measured/computed online for a sequence of batches, plus values of the 16 quality attributes determined in the lab at the end of the same batches. Two separate datasets are available for the analysis. The first dataset covers two and a half months of stable operation of the process, comprising 190 batches from six bioreactors (BR7 was under maintenance at that time, therefore not operating). This dataset comprise data collected before the development of the decreasing quality trend reported in Figure 3.3 and is referred to as "stable dataset" in this Chapter. The second dataset spans over three months of operation and captures the quality trends reported in Figure 3.3. On the timespan of the second dataset, referred to as "trend dataset" in this Chapter, all seven bioreactors were operating. A total of 184 batches is found in this dataset. The number of batches for each bioreactor in each one of the datasets is represented in Figure 3.4.

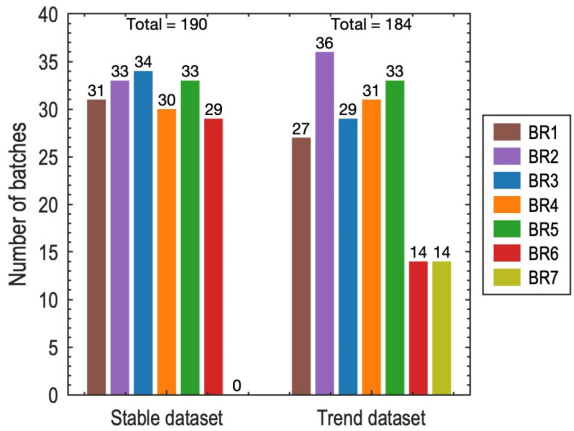


Figure 3.4. *Number of batches for each bioreactor in the two available datasets.*

The available datasets are used to carry out a comprehensive investigation of the bioconversion process. The stable dataset is used to assess potential differences among the performance of the bioreactors in the array in terms of both online and offline variables. The trend dataset is used to investigate the quality trends and to suggest corrective actions. The data analytics methods used to accomplish these tasks are described in the next Section.

3.3 Process understanding by latent-variable modeling

The aim of this Section is to recall the fundamentals of the data analytics methods used for the study described in this Chapter. The fundamental rationale of their application as to achieve the objectives stated in the previous Section is discussed as well.

3.3.1 Principal component analysis for data exploration

PCA, the rationale of which has been introduced in Section 2.1, is used in this Chapter to assess potential differences among the seven bioreactors. Note that a simple exploratory analysis is carried out, rather than a proper discriminant analysis, the latter method being recommended by Louwerse et al. (1999b). In fact, the objective of the study carried out here is simply to verify the presence of grouping in the score plot induced by the correlation structures of data from different bioreactors, rather than to identify class separation boundaries. Therefore, using PCA to assess differences among bioreactors is considered appropriate in this study.

The stable dataset is used for PCA modeling. Differences among bioreactors are sought after according to both process (online) and product quality (offline) data. PCA is applied to process data after the preprocessing operations describes in Section 3.4.1 to obtain matrix $\mathbf{X}$, while it is applied to the matrix of product quality data directly. In both cases, the score plot is visually assessed to search for clusters of scores representing batches from different bioreactors. If any clustering is spotted, loadings are used to diagnose the differences among scores in the clusters.

3.3.2 Joint-Y partial least-squares regression interpretation and inversion

JYPLS (see Section 2.4) is used to develop a regression model between process variables, processed as described in Section 3.5.1 to obtain the matrix sequence $\{\mathbf{X}_1, \dots, \mathbf{X}_P\}$[4], and a selected subset of variables describing the product quality at the end of each batch, arranged as the matrix sequence $\{\mathbf{Y}_1, \dots, \mathbf{Y}_P\}$. The JYPLS model is interpretable as a standard PLS model (García-Muñoz et al., 2005), therefore regression coefficients are investigated (Burnham et al., 2001) to search for the likely cause of the decreasing quality trend shown in Figure 3.3.

The interpretation of JYPLS regression coefficients can offer guidance to recover from the decreasing quality trend according to a univariate perspective. However, a multivariate approach offers significant advantages, including the preservation of the correlation structure among process variables. To this end, the LVMI approach, introduced in Section 2.5, is used to understand the difference between batches yielding high-quality and low-quality products. DI is selected for inversion by virtue of its computational simplicity.

3.4 Assessment of differences among bioreactors

In this Section, PCA is used to analyze the stable dataset. The objective is to seek for potential differences among the six bioreactors operating in parallel at the time of acquisition of the data at hand. Both process and product quality data are used for the analysis. MATLAB R2022a (The Mathworks, 2022a) is used for all the computations (with in-house-developed code).

[4] Recall that the subscript *P* stands for "plant" in the general terminology used in JYPLS literature, but "units" or "blocks" have equivalent meanings.

3.4.1 Analysis of online data

Results concerning online data are described first. Preliminary analyses highlighted that two of the online variables, the volume and the water fraction in the off-gas, feature a highly unstructured variability that can severely bias the models. Therefore, these two variables are removed from the dataset. The variables considered in the analysis described hereby are reported in Table 3.1.

Table 3.1. *Process variables considered in the PCA model of online data.*

ID	Variable	Symbol
X01	Temperature	T
X02	Pressure	P
X03	pH	pH
X04	DO	DO
X05	Mixing power consumption	W
X06	Substrate feed flow rate	$\dot{V}_{\mathrm{S}}$
X07	Air feed flow rate	$\dot{V}_{\mathrm{air}}$
X08	pH buffer 1 feed flow rate	$\dot{V}_{\mathrm{pHB1}}$
X09	pH buffer 2 feed flow rate	$\dot{V}_{\mathrm{pHB2}}$
X10	Additive feed flow rate	$\dot{V}_{\mathrm{A}}$
X11	Substrate added volume	V_{S}
X12	Air added volume	V_{air}
X13	pH buffer 1 added volume	V_{pHB1}
X14	pH buffer 2 added volume	V_{pHB2}
X15	Additive added volume	V_{A}
X16	Off-gas O_2 fraction	$y_{\mathrm{O_2}}^{\mathrm{out}}$
X17	Off-gas CO_2 fraction	$y_{\mathrm{N_2}}^{\mathrm{out}}$
X18	Off-gas N_2 fraction	$y_{\mathrm{CO_2}}^{\mathrm{out}}$
X19	OUR	OUR
X20	CER	CER

The data are processed by the synchronization-unfolding approach to batch data analytics described in Section 2.6.1. Profiles are reasonably aligned, and synchronization is deemed to not be needed. However, the number of observations per batch varies due to the content of the bioreactor being withheld for some time after batch completion if the buffer tank to which the bioreactors are withdrawn to is not empty. Such “tails” are not considered as parts of the batch

and are assumed to not provide any additional information. Therefore, the number of observations per batch is made even by truncation (Rothwell et al., 1998). This operation results in a sequence of 190 synchronized matrices containing profiles of 20 variables recorded on 200 time samples each. The synchronized matrices are transformed by BWU (Wold et al., 1987b) to obtain matrix **X**, containing 190 rows and $20 \times 200 = 4000$ pseudo-variables (each row represents a batch and contains profiles of all process variables concatenated horizontally). The **X** matrix is autoscaled and analyzed by PCA. Note that **X** contains data from all bioreactors, hence the autoscaling considers mean and variance of each pseudo-variable on the whole array of bioreactors, thus preserving relative relationships among data from different bioreactors. Scores from the PCA model on the first four PCs are shown in Figure 3.5. Recall that each dot in the figure represents a complete batch.

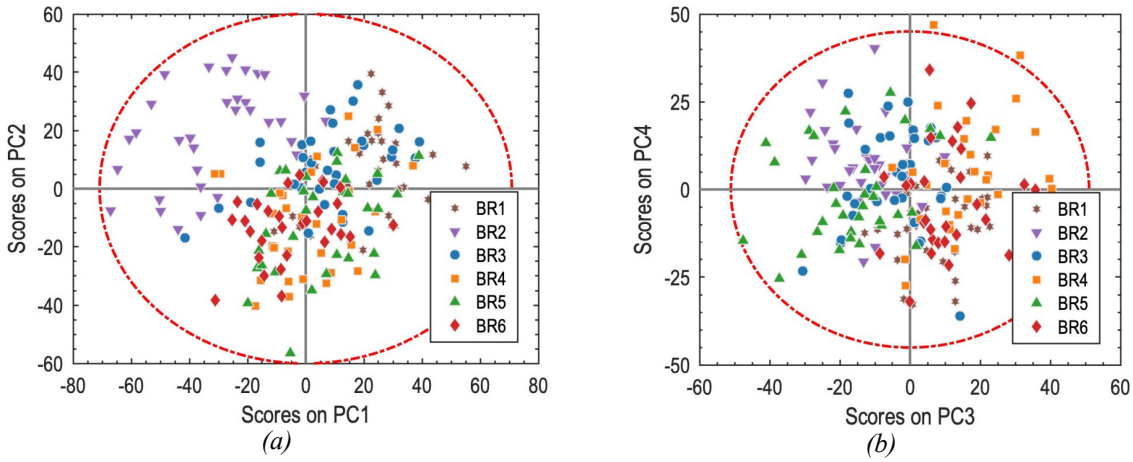


Figure 3.5. *Scores of the PCA model of process variables in the stable dataset: (a) first and second PCs; (b) third and fourth PCs.*

Scores of different bioreactors show a clear separation in Figure 3.5, uncovering different behaviors of the bioreactors in the array. BR2 shows the most defined separation, denoting a strong difference with respect to other bioreactors; however, also other units show some degree of separation, for example BR3 and BR6. Only BR2 will be discussed herein for the sake of brevity. Figure 3.5(a) clearly shows that BR2 forms a separate cluster with respect to other bioreactors, featuring negative scores on the first PC and positive scores on the second PC. Separation from clusters of other bioreactors is seen also in Figure 3.5(b), where BR2 shows negative scores on the third PC and positive scores on the fourth PC.

The loadings of the PCA model, shown in Figure 3.6 can be interpreted to gain insight on the causes of the grouping. In general, loadings represent a "map" for the score plot. Specifically, if a score has a positive (negative) value on a given PC, variables with positive loadings on that PC have a high (low) value in the observation represented by the score, while variables with negative loadings have low (high) values. However, given the clustering detected in Figure 3.5, the loadings describe differences between BR2 and the other bioreactors. Scores of BR2 are

mostly negative on the first and third PCs, while they are mostly positive on the second and fourth PCs. Therefore, when compared to the other bioreactors, BR2 features higher values of the variables with negative loadings on the first and third PCs, or positive loadings on the second and fourth PCs, and *vice-versa*. Recall that, in BWU, one loading is obtained for each time sample in the profile of each variable, hence the above considerations hold true for each individual time instant. Variable identifiers used in Figure 3.6 are reported in Table 3.1.

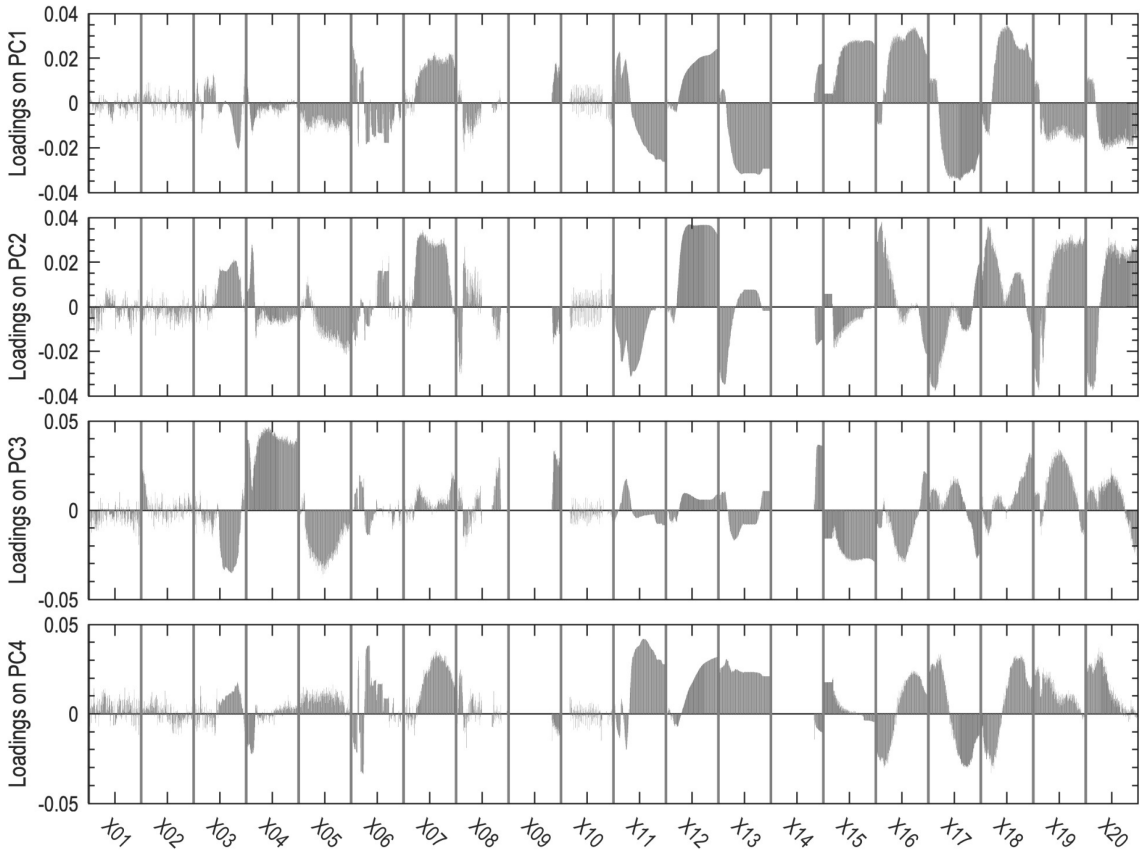


Figure 3.6. *Time profiles of loadings on the first four PCs of the PCA model for each process variable in the stable dataset.*

Focusing on the first PC, Figure 3.6 reveals that OUR and CER (variables X19 and X20, respectively) show positive loadings in the biomass growth phase of the batch, while loadings are negative in the production phase. The patterns of loadings of OUR and CER are reversed for the second PC. These patterns hint to the fact that, when compared to the other bioreactors, BR2 has a lower respiration rate in the growth phase, while the rate is higher in the production phase. These conclusions are confirmed by fractions of oxygen and carbon dioxide in the off-gas (variables X16 and X17, respectively), the former being lower and the latter higher in BR2 than in other units. However, also the off-gas nitrogen fraction (variable X18) is lower: this hints to a better gas-to-liquid mass transfer taking place in BR2, simplifying the control the DO. The loadings on the third PC in Figure 3.6 also show that the DO (variable X04) in BR2 is consistently lower (positive loadings) than in other bioreactors. This effect is due to the additive

(variable X15), the loadings of which are negative on the same PC, thus revealing that BR2 receives a smaller volume of additive than other units. The additive used in this process can impact the mass transfer properties of the system. A possible explanation for the lower DO is that the mass transfer properties of BR2 lead to a better utilization of the DO by the microorganisms. However, one should note that the loadings of the additive added volume on the first PC in Figure 3.6 are positive, meaning that BR2 receives a larger volume of additive when compared to other bioreactors. The inconsistency between the interpretations of the first and third PCs is investigated by visual analysis of the raw profiles, which reveals that adding a large volume of additive actually causes the DO to decrease significantly, while no effect on the DO is found when the added volume of additive is below a given threshold.

While respiration is doubtlessly a critical factor, BR2 features a different pattern also in the pH profiles when compared to other bioreactors. Loadings on pH (variable X03) on the second and third PCs show "hills" in the central part of the profiles, positive in the former PC and negative in the latter PC. This hints to the fact that the pH is on average higher in BR2 then in other units during the pH shift shown in Figure 3.2(b), which could be due to two reasons: the shift occurs later, and the transition to the new pH set-point is slower. However, the loadings on the added volume of the pH buffer 1 (variable 13) are negative on the first PC and positive on the fourth PC, which means that the pH control in BR2 requires a larger volume than other bioreactors (pH buffer 1 is alkaline). This uncovers the fact that the different respiration pattern in BR2 could induce a tendency in the microorganisms to lower the pH in the earlier phase of the batch, thus the larger volume of pH buffer 1 needed, which turns out to be beneficial to the pH control in the later phase. A possible explanation is that the pH buffer 1 has an effect on the metabolism of the microorganism. This conjecture if further explored in the analysis of data related to the product quality.

3.4.2 Analysis of offline data

PCA is applied to the product quality data in the stable dataset directly. Matrix **X** contains observations of the 16 product quality variables recorded at the end of the 190 batches analyzed in the previous Section. The variables considered in the analysis are reported in Table 3.2.

PCA is applied to the matrix **X** after autoscaling. Similarly to the analysis of process variables, observations in **X** come from all the six bioreactors operating at the time of acquisition of the stable dataset. Therefore, autoscaling preserves the relative relationships among bioreactors. The interpretation of the loadings follows the same rationale described in the previous Section. Scores and loadings of the PCA model are reported in Figure 3.7. The labels of variables in Figure 3.7(b) are reported in Table 3.2.

Figure 3.7(a) highlights a well-defined separation of scores of BR2 from scores of other bioreactors. The separation develops along the first PC, hence only loadings on such a PC are reported in Figure 3.7(b). Loadings highlight that BR2 yields higher productivity and biomass

Table 3.2. *Product quality variables considered in the PCA model of offline data.*

ID	Variable	Symbol
Y01	BDO concentration	c_{BDO}
Y02	Productivity	Pr
Y03	Yield	Y
Y04	Biomass concentration	OD
Y05	Quality index	I
Y06	Residual substrate concentration	c_{S}
Y07	Byproduct 1 concentration	c_{B1}
Y08	Byproduct 2 concentration	c_{B2}
Y09	Byproduct 3 concentration	c_{B3}
Y10	Ionic species 1 concentration	c_{I1}
Y11	Ionic species 2 concentration	c_{I2}
Y12	Ionic species 3 concentration	c_{I3}
Y13	Ionic species 4 concentration	c_{I4}
Y14	Ionic species 5 concentration	c_{I5}
Y15	Ionic species 6 concentration	c_{I6}
Y16	Ionic species 7 concentration	c_{I7}

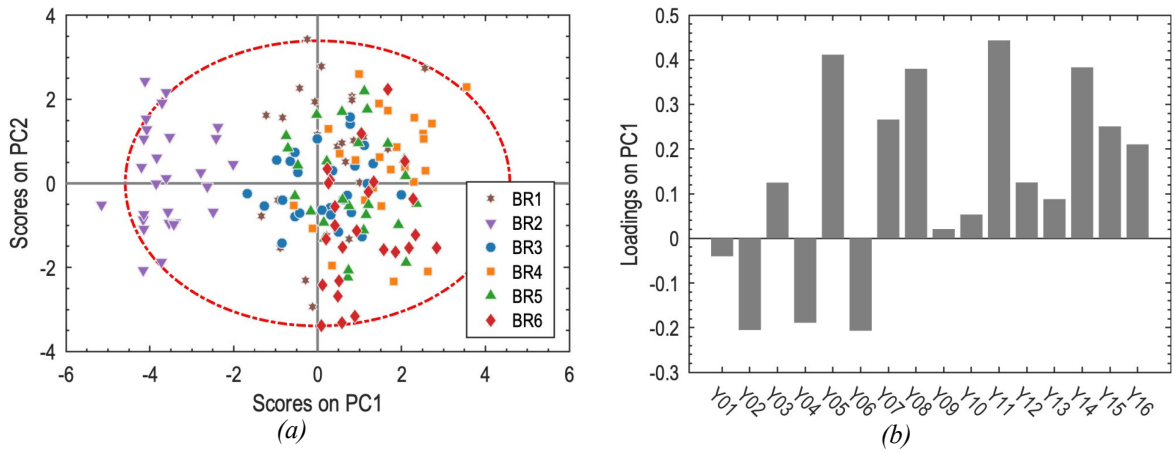


Figure 3.7. *(a) Scores and (b) loadings of the PCA model of product quality variables in the stable dataset.*

concentration (variables Y02 and Y04, respectively) compared to other bioreactors, even though the yield (variable Y03) is slightly lower. The final BDO concentration (variable Y01) does not significantly differ from other bioreactors.

The most prominent difference is found in the quality index (variable Y05) and ionic species (variables Y10 to Y16). All these variables feature positive loadings in Figure 3.7(b), which highlights that the solution processed by BR2 has lower quality index and concentration of ionic

species than other bioreactors. I1 (variable Y10) is particularly interesting. As mentioned in Section 3.2.3, I1 is introduced in the bioreactor with pH buffer 1. The analysis of online data highlights that BR2 requires a higher volume of pH buffer 1 than other bioreactors to properly control the pH. However, the final concentration of I1 is not larger than in other bioreactors. This supports the conjecture that pH buffer 1 has an effect on the metabolism of the microorganism, as the I1 not found in the solution must be within the cells. The metabolism of the microorganisms in BR2 may be slightly different overall when compared to other units, a point supported by the fact that the solution processed by BR2 shows a different distribution of ionic species uniquely introduced in the bioreactors with the culture medium (variables Y12 to Y16), despite the culture medium being the same for all bioreactors. Figure 3.7(b) also highlights that BR2 tends to yield less byproducts (variables Y07 to Y09) than other bioreactors. Such byproducts could cause the pH of the solution to decrease, which justifies the slower decrease of pH in the set-point change in BR2 discussed in the previous Section.

In summary, BR2 shows better performance with respect to the other bioreactors in the array. The differences in OUR, CER, and off-gas composition indicate that BR2 has a different respiration pattern. This is also related to the lower volume of additive, a substance affecting the gas-to-liquid mass transfer, required by BR2 when compared to other bioreactors. The smaller additive volume improves the mass transfer and simplifies the control of the DO, which can be kept closer to the optimal level during the production phase, minimizing the byproducts obtained in the bioconversion. The byproducts tend to decrease the pH of the solution, hence a lower concentration of such species makes the control of the pH easier and slows down the pH shift when the set-point change kicks in. However, the different respiration pattern of the microorganisms in BR2 requires more pH buffer 1 in the early phase of the batch, with respect to other units. Therefore, one may conjecture that, in BR2, there is a high production rate of byproducts in the biomass growth phase, but, since the biomass concentration is lower in this phase, less byproducts are formed overall. Finally, pH buffer 1 is found to influence the metabolism of microorganisms, which may contribute to lower the formation of byproducts in the later phase of the batch.

After discussion with the plant personnel, these findings are attributed to the fact that BR2 features a different impeller with respect to other bioreactors. While this difference was deemed to not be relevant during the construction of the plant, the PCA-based analyses described in this Section prove otherwise.

3.5 Troubleshooting of the decreasing trend in end-of-batch quality

In this Section, the calibration of the JYPLS model is described, elucidating the motivations that lead to the choice of this sophisticated modeling approach over simpler methods. The interpretation of the model, aimed at understanding the likely causes of the decreasing trend in

product quality, is described as well. Finally, a proper multivariate approach (LVMI) is adopted to confirm the findings of model interpretation. The trend dataset is used for all the aforementioned analyses. All the computations are carried out on MATLAB R2022a (The Mathworks, 2022a) with in-house-developed code.

3.5.1 JYPLS model calibration and assessment

The outcomes of the analyses described in the Section 3.4 support the hypothesis that a single predictive model for the whole array of reactors may not be appropriate to investigate the decreasing trend in product quality shown in Figure 3.3. However, the quality trend seems to affect all the bioreactors. Therefore, we aim at modeling common phenomena tanking place in all the units, hence developing one model for each bioreactor would not be appropriate either. Therefore, we resort to JYPLS by virtue of its ability to model arrays of parallel units extracting LVs describing both general phenomena common to the whole array (between-unit correlation) and phenomena characteristic of each single unit (within-unit correlation). In particular, the former correlation is set to take place in the quality space due to the JYPLS model principles (García-Muñoz et al., 2005), which is the space of interest in the analysis presented herein.

Preliminary evaluations highlighted that process variables of three of the bioreactors (BR1, BR2, and BR7) show highly unstructured variability over the timespan covered by the trend dataset, which masks the input-output relationship. For these units, unacceptable validation performances are obtained with both JYPLS and single PLS models. Therefore, these three bioreactors are not considered in the analysis described in this Section. Only batches manufactured in BR3, BR4, BR5, and BR6 are included in the dataset.

The data for JYPLS modeling are arranged as two sequences of matrices: $\{\mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5, \mathbf{X}_6\}$ and $\{\mathbf{Y}_3, \mathbf{Y}_4, \mathbf{Y}_5, \mathbf{Y}_6\}$. Matrix $\mathbf{X}_p$ is a BWU matrix containing batches from bioreactor BRp. The batch data from bioreactor BRp are pre-processed as described in Section 3.4.1. The volume and the water fraction in the off-gas are neglected for the same reasons outlined in the aforementioned Section. For ease of interpretation, time profiles of all four bioreactors are truncated at the same number of time samples: 202. The same 20 process variables listed in Table 3.1 are considered for all bioreactors, yielding 4040 pseudo-variables in matrix $\mathbf{X}_p$. The numbers of observations for each bioreactor are reported in Figure 3.4. Matrix $\mathbf{Y}_p$ contains measurements of the end-of-batch quality variables of bioreactor BRp. Only 2 variables describing the product quality at the end of each batch are considered for JYPLS modeling: the concentration of BDO and the quality index (respectively variables Y01 and Y05 in Table 3.2), as they are the variables showing the most defined trends, besides being the main quality variables for the process.

The JYPLS model is calibrated following the procedure outlined by García-Muñoz et al. (2005). The number of LVs is selected by leave-one-out cross-validation (Facco et al., 2014, 2020; Meneghetti et al., 2012; Rudnitskaya et al., 2017). Different numbers of LVs yield the best performances for different bioreactors. However, the optimal compromise is found to be $A = 4$

LVs. The best performances in cross-validation are achieved for BR5 and are reported in Table 3.3 in terms of determination coefficients of the two output variables.

Table 3.3. *Determination coefficients of the JYPLS model in cross-validation for BR5.*

LV	R^2 on c_{BDO}	R^2 on I
1	0.6556	0.1902
2	0.6718	0.3082
3	0.6630	0.3892
4	0.6870	0.4211
5	0.7423	0.4226
6	0.7365	0.4359
7	0.7586	0.4349
8	0.7537	0.4248
9	0.7543	0.4253

Table 3.3 highlights that using 6 or 7 LVs would yield better a performance for BR5. However, validation performances of other bioreactors severely degrade after $A = 4$, denoting the onset of overfitting. The determination coefficient for the concentration of BDO is $R^2_{c_{BDO}} = 0.6870$, while the one for the quality index is $R^2_I = 0.4221$. Both coefficients are deemed to be satisfactory considering the low signal-to-noise ratio of the data. However, one must be careful regarding the outcomes of the model interpretation for the quality index, given the low determination coefficient for this output variable.

3.5.2 Understanding the quality trend by model interpretation

In this Section, the JYPLS model is interpreted to gain insight on the likely causes of the decreasing trend in end-of-batch product quality. The JYPLS scores for all the considered bioreactors and joint-Y loadings on the first two LVs are reported in Figure 3.8. The scores are colored by BDO concentration scaled for confidentiality reasons.

Figure 3.8(a) shows that the model clearly captures the decreasing quality trend: scores move from the positive side to the negative side of the first LV as the concentration of BDO decreases over the timespan of the dataset. Furthermore, the top panel of Figure 3.8(b) highlights that the trends of BDO concentration and quality index are correlated, the latter variable increasing as the former one decreases (see Figure 3.3). Therefore, the movement of the scores along the first LV describes both quality variables. However, the second LV mostly models the quality index alone. Therefore, while the two quality variables are correlated to some extent, the quality index is affected by other factors not included in the data at hand. This also explains the scarce cross-validation performance of JYPLS on the quality index shown in Table 3.3.

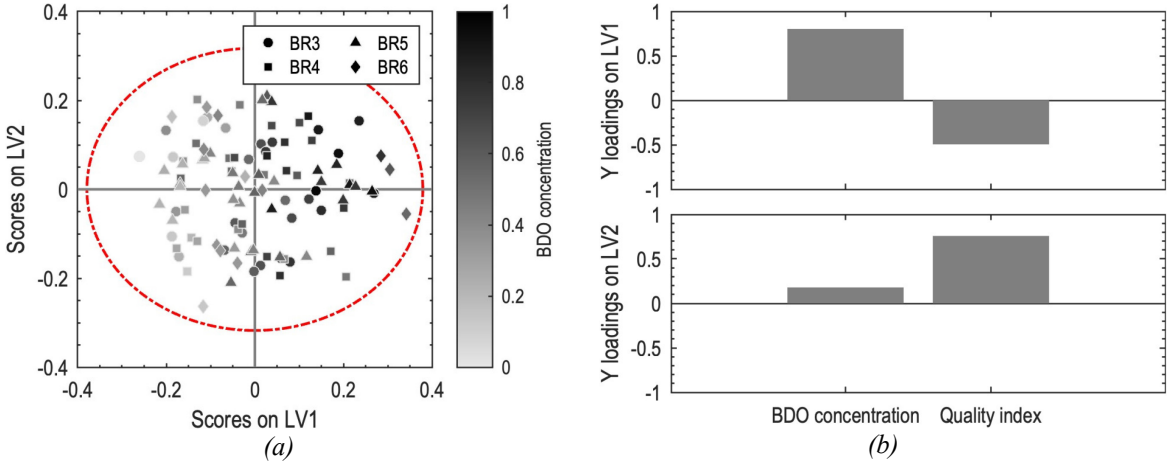


Figure 3.8. *(a) Scores and (b) joint-Y loadings of the JYPLS model developed on data from the trend dataset. The scores are colored by BDO concentration scaled between 0 (low concentration) and 1 (high concentration).*

For the sake of brevity, model interpretation is discussed for BR5 only, being the best-modeled unit in the array. The interpretation is carried out on the basis of the JYPLS outer regression coefficients, as they give a quantitative measure of the input-output relationship. In particular, a positive regression coefficient implies that a high value of the input variable yields a high value of the output variable, while a negative regression coefficient implies the reverse; furthermore, the higher the absolute value of a coefficient, the larger the variation of the output corresponding to a given variation in the relevant input. The coefficients for BR5 are reported in Figure 3.9. Note that, as the $\mathbf{X}_p$ matrices are obtained by BWU from batch data, one coefficient is obtained for each time sample in the profile of each process variable.

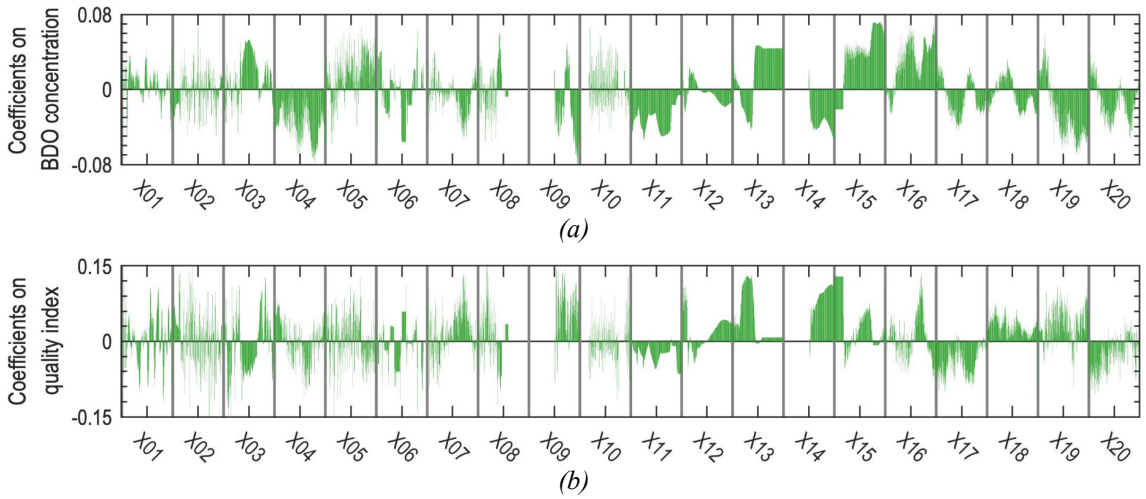


Figure 3.9. *Time profiles of JYPLS outer regression coefficients for each process variable on (a) BDO concentration and (b) quality index for BR5.*

Regression coefficient for the end-of-batch BDO concentration, shown in Figure 3.9(a), are discussed first. The DO (variable X04) shows consistently negative regression coefficients, hinting to the fact that high values of DO yield a low BDO concentration. This finding matches

the available information of the metabolism of microorganisms and the outcomes of the analysis of the stable dataset discussed in Section 3.4. The coefficients of OUR and CER (variables X19 and X20, respectively) confirm that a high respiration rate yields a low BDO concentration. A closer analysis of the raw profiles of these three variables reveals a drift of all of them towards higher values in the timespan covered by the data. The drift in DO causes the process to stray from the optimal conditions, inducing a high respiration rate in the microorganisms, hence the increased OUR and CER. The suboptimal DO also causes a less efficient utilization of the substrate, as denoted by the negative coefficients of its added volume (variable X11). This leads to the formation of more byproducts, which tend to lower the pH, hence to increase the demand of pH buffer 2 volume (variable X14), its coefficients being negative (pH buffer 2 is alkaline). The increased demand of pH buffer 2 is also related to the pH profile itself. In Figure 3.9(a), the coefficients on pH (variable X03) show a "hill" in the central part of the profile, hinting to the fact that shifting the pH to the low set-point later in the batch yields a higher BDO concentration. This can be achieved by adding more pH buffer 1 (variable X13) at the beginning of the batch and overall, hence the pattern shown by the coefficients of such variable in Figure 3.9(a). However, an analysis of the raw profiles of pH in the dataset shows that the duration of the pH "high shelf" steadily decreased over the timespan of the data.

After discussions with the plant personnel, we attribute the earlier pH set-point shift to a deliberate action performed by plant operators. The coefficients on the quality index in Figure 3.9(b) show that a large added volume of pH buffer 1 (variable X13) is responsible for a high end-of-batch quality index. In fact, throughout the timespan of the trend dataset, plant operators tried to decrease the end-of-batch quality index by limiting the added volume of pH buffer 1 and triggering the pH set-point shift as soon as the preset maximum volume is achieved. This action is guided by a univariate reasoning (the effect of a single variables is considered). However, the multivariate model-based analysis discussed here proves that such an action is detrimental to both the BDO concentration and the quality index, due to the pH shift happening earlier and causing, jointly with the DO drift, the formation of more byproducts. The latter point also implies a high demand of pH buffer 2 (variable X14) to control the pH, hence an increase in the quality index due to a large added volume of this chemical, as can be inferred by the positive coefficients of this variable in Figure 3.9(b).

Figure 3.9(b) also highlights that adding a large volume of additive (variable X15) at the beginning of the batch contributes to increasing the end-of-batch quality index, as denoted by the positive coefficients. In fact, the additive has a remarkable effect on the gas-to-liquid mass transfer, as discussed in Section 3.4, which may in turn influence the growth phase of the microorganisms by affecting the DO, thus compromising the whole batch. This conclusion is supported by Figure 3.9(a) as well, where the coefficients of the additive added volume are negative in the earlier phase of the batch but become strongly positive in the later phase. This pattern suggests that little additive should be introduced in the bioreactor before the batch start

as to ease the control of the DO in the biomass growth phase, while a larger volume should be used in the production phase to control the mass transfer properties of the system, thus the DO. Other minor conclusions can be drawn exploring the coefficients in Figure 3.9. However, they are not discussed here for brevity. Finally, qualitatively similar conclusions can be drawn concerning the other bioreactors, even though the differences in the process units cause slight variations in the interpretation, most notably for the DO-additive interaction.

3.5.3 Guidelines for process recovery by model inversion

The model interpretation discussed in the previous Section offers precious guidelines to recover the quality loss experienced in the plant in the timespan of the trend dataset. Such an analysis also makes clear that both BDO concentration and the quality index are not influenced by single variables independently: complex interactions between the phenomena underlying the process exist due to the multivariate nature of the system under investigation. Therefore, the conclusions drawn from the JYPLS model interpretation (based on the analysis of one or few variables at a time) must be verified by a proper multivariate approach: LVMI is used to this end. The outcomes of LVMI also pave the way for future, data-driven process improvement.

Two targets for the product quality are set for LVMI:

- a high-quality target denoted as "HQ" and set as $\mathbf{y}_{\text{des}}^{\text{HQ}} = [0.874 \quad 0.136]^{\text{T}}$;
- a low-quality target denoted as "LQ" and set as $\mathbf{y}_{\text{des}}^{\text{LQ}} = [0.241 \quad 0.768]^{\text{T}}$.

In both cases, the first component refers to the scaled BDO concentration, while the second component is the scaled quality index. $\mathbf{y}_{\text{des}}^{\text{HQ}}$ represents a product with high BDO concentration and low quality index, while $\mathbf{y}_{\text{des}}^{\text{LQ}}$ identifies the opposite, undesirable situation. The two targets are visualized in the product quality space of the historical data in Figure 3.10. The points representing batches in the historical dataset are colored by batch number, the scale of which corresponds to the abscissa of the data reported in Figure 3.3.

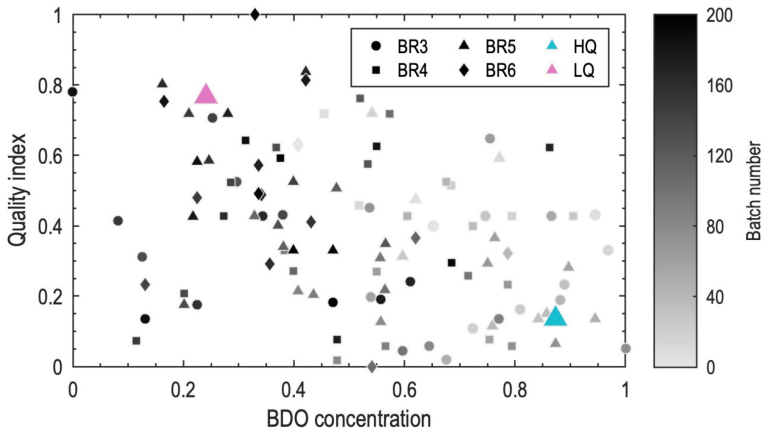


Figure 3.10. *High-quality (HQ) target and low-quality (LQ) targets shown in the product quality space and compared to the quality of products manufactured in the historical batches in the dataset. The points representing batches in the historical dataset are colored by batch number corresponding to the abscissa of Figure 3.3.*

The two quality targets are used in the inversion of the JYPLS model. DI is selected to perform the inversion by virtue of its simplicity and computational efficiency. This choice is motivated by the simple confirmatory nature of the LVMI. For the same reason, only the particular solution to DI, the $\mathbf{t}_{\text{des,p}}$ defined in (2.77), is considered and the two-dimensional null space arising from the inversion (recall that $A = 4$ and $V_Y = 2$ in this case) is disregarded. The resulting profiles of process variables are visualized to confirm the outcomes of the model interpretation outlined in Section 3.5.2. While the inversion of the JYPLS model could be used to design process conditions for any of the bioreactors considered in the model, only BR5 is discussed here for the same reasons explained in the previous Section.

The scores obtained by DI of the two quality targets are reported in Figure 3.11, together with the designed profiles of two selected process variables, DO and pH, for BR5. The scores of batches in the historical dataset are colored by scaled BDO concentration.

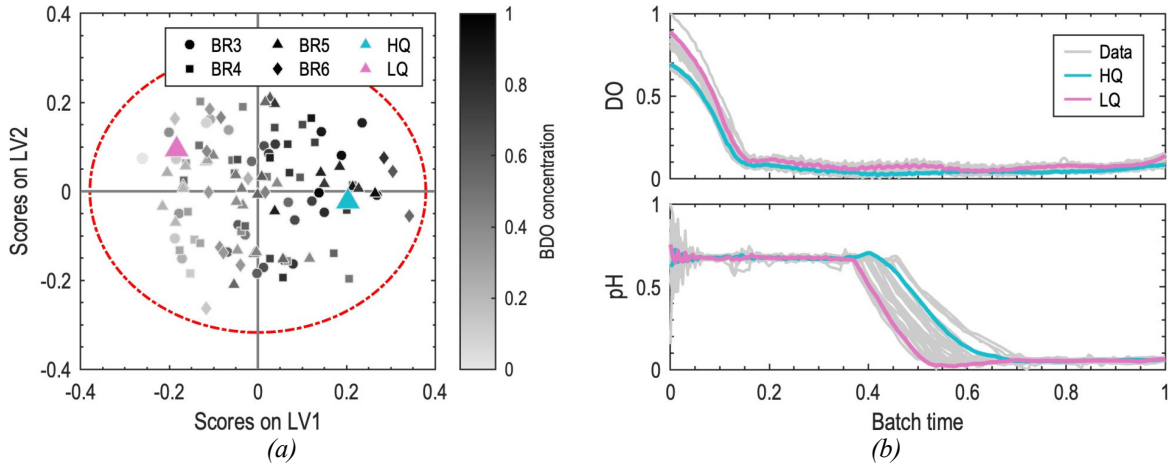


Figure 3.11. *(a) Scores obtained by DI of the high-quality (HQ) and low-quality (LQ) targets compared to scores of batches in the historical dataset, colored by BDO concentration scaled between 0 (low concentration) and 1 (high concentration), and (b) designed profiles of two selected process variables compared to the historical data of BR5.*

The scores in Figure 3.11(a) show that $\mathbf{y}_{\text{des}}^{\text{HQ}}$ and $\mathbf{y}_{\text{des}}^{\text{LQ}}$ are projected onto the space of joint LVs in a meaningful way, confirming that the model explains the quality trend. Furthermore, both the targets are projected within the model validity region (the confidence ellipse), hence results of the inversion are deemed reliable.

Figure 3.11(b) confirms the main points outlined in Section 3.5.2. The inversion of the low-quality target yields a DO profile with a consistently higher value for the whole production phase, when compared to the results of the inversion of the high-quality target. The designed pH profiles clearly illustrate the detrimental effects of anticipating the shift in the pH set-point, with the profile obtained from $\mathbf{y}_{\text{des}}^{\text{LQ}}$ initiating the shift as early as allowed by the data at hand and proceeding to the new set-point with a steep ramp. Conversely, the profile obtained from $\mathbf{y}_{\text{des}}^{\text{HQ}}$ keeps a high level of pH for a longer period and reaches the new set-point with a softer ramp. Profiles of other process variables (not shown for conciseness) confirm these points.

The outcomes of LVMI provide strong evidence on the likely causes of the decreasing trend of the end-of-batch product quality and offers guidelines for future data-driven process improvement. However, we cannot provide any experimental proof of the effectiveness of such guidelines. In fact, major changes to the equipment and structure of the upstream process took place in the plant shortly after the completion of this study. The motivation for such changes was unrelated to the problem we investigated and affected the reference values of the quality variables, thus preventing us from verifying the resolution of the issue we investigated by testing the proposed guidelines on the plant. We strongly believe in the value of our approach nonetheless, and we are confident of its effectiveness in light of the strong conclusions we put forward by data analytics.

3.6 Conclusions

In this Chapter, we carried out a comprehensive analysis of the bioconversion step in the upstream process of an industrial biorefinery for BDO production. The process features seven bioreactors operating in cycled fed-batch regime as to continuously feed the downstream process of the biorefinery. We addressed a decreasing trend in the end-of-batch product quality affecting all bioreactors, the main effects of which manifested as a gradual decrease in the final BDO concentration and an increase in a process-specific quality index, a fundamental variable for the proper operation of the downstream process following the bioreactors.

We applied PCA to two and a half months of data recorded during stable operation of the process to assess differences among the bioreactors. The analysis highlighted that one of the bioreactors yields significantly better performances with respect to the other ones. The properties of PCA allowed us to diagnose the main differences between the identified bioreactors and the other units in terms of both process variables and product quality attributes.

We resorted to JYPLS to address the decreasing trend in end-of-batch product quality, as captured by three months of data. The choice of JYPLS was motivated by its ability to model arrays of parallel units considering both within-unit and between-unit correlations. Model interpretation allowed us to attribute the decreasing trend in the BDO concentration to a drift of the DO towards higher values in the production phase of the batch, which lead to an increase in the respiration rate and to a less efficient utilization of the substrate. Furthermore, a reduction in the added volume of one of the pH buffers, an action gradually implemented in the plant over the timespan of the dataset as an effort to reduce the quality index, caused an early onset of the pH set-point shift, which was demonstrated to be detrimental to both the BDO concentration and the quality index itself. We confirmed all the conclusions drawn from model interpretation by LVMI, specifically by DI of two quality targets: one for a high-quality product and one for a low-quality product. The profiles obtained by inversion of the two quality targets provided clear guidelines to identify the best process conditions and can be used to recover from the

undesirable quality loss experienced by the plant. However, we could not validate our results experimentally due to significant changes taking place in the plant set-up simultaneously to the completion of this study (unrelated to the problem we investigated). We are nonetheless confident in the effectiveness of our approach.

This study provides strong evidence of the value of a data-driven approach to process understanding and improvement in challenging industrial scenarios. The latent-variable models selected allowed to: gain a remarkable knowledge on the process and on the fundamental physical and biological phenomena taking place in the bioreactors; model an array of parallel bioreactors under limited data availability for each unit; address the decreasing trend in the end-of-batch product quality; develop guidelines for future process improvement. The fact that these results were obtained in a one-of-a-kind process implementing cutting-edge technology further supports the value of the Industry 4.0 approach adopted in this Thesis.

Chapter 4

Hybrid model-based monitoring of a membrane separation process[5]

This Chapter discusses the challenges of membrane fouling monitoring in biorefineries and proposes a hybrid modeling strategy to characterize reversible and irreversible fouling in multi-module membrane separation systems. While the typical approach relies on the limited insight offered by the average resistance of multi-module systems, the proposed strategy combines PLS modeling and Darcy's equation to estimate individual resistances of all membrane modules. Monitoring individual resistances provides valuable insight into the fouling state of the membranes, offering advantages over monitoring trans-membrane pressures and permeate fluxes in terms of data variability, process changes, module interactions, and fouling dynamics.

4.1 Introduction

A typical feature of biorefineries based on bioconversion is that the mixtures processed in bioreactors, containing the desired products, are usually diluted solutions, which entails high downstream processing costs (Bähner et al., 2021; Cuellar et al., 2020; Martín et al., 2013). Membrane filtration has been identified as an effective technology to remove cells (and high molecular weight compounds) from the solutions containing the main product (Prochaska et al., 2018; Shimizu et al., 1993). The topic has been widely investigated: membrane operations are in fact becoming increasingly relevant in biorefineries (Abels et al., 2013; Carstensen et al., 2012; Ennaceri et al., 2022; Gerardo et al., 2014; Saha et al., 2017) due to their better scalability and lower operating costs compared to conventional thermal separation processes (Ennaceri et al., 2022; Gerardo et al., 2014; Jiang et al., 2013; Saha et al., 2017).

Among the membrane-based operations, pressure-driven membrane separation processes, for example ultrafiltration and nanofiltration, are the most used ones to separate biomass from the bioconversion products (Rudolph et al., 2019). However such processes can suffer from membrane fouling (Arnese-Feffin et al., 2023c; Mancini et al., 2020; Prochaska et al., 2018). Membrane fouling can be characterized as reversible or irreversible: the former is relatively fast, triggers short-term process disruption, and can be removed by hydraulic or chemical cleaning; the latter acts slowly and causes long-term membrane degradation (Huang et al., 2021;

[5] Part of the research discussed in this Chapter has been published as a journal paper (Arnese-Feffin et al., 2024).

Shi et al., 2014). The two fouling types influence one another and affect membranes at the same time, causing decrease of permeate flux in constant pressure filtration (Abels et al., 2013) or increase of TMP (pressure difference across the membrane) in constant flux separation (Klimkiewicz et al., 2016), also implying an increase in energy expenditure in the latter case. Fouling, in particular the reversible one, typically causes membrane separation processes filtering the outlet of bioreactors to run in semi-continuous regime, meaning alternating operation and cleaning phases (Klimkiewicz et al., 2016). The effect of membrane fouling on a sequence of operation and cleaning cycles is schematically depicted in Figure 4.1.

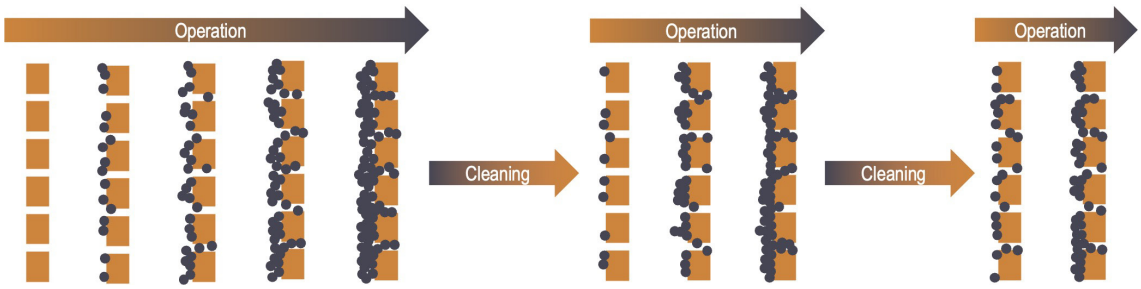


Figure 4.1. *Action of membrane fouling on a sequence of operation-cleaning cycles. Reversible fouling is caused by material accumulating on the membrane (gray dots) as the filtration proceeds. Mechanical/chemical cleaning can remove most of the accumulated material, but a fraction contributes to irreversible fouling and long-term degradation of the membrane.*

Prompt monitoring of fouling is vital for efficient operation of membrane processes. This task can be accomplished experimentally at the processing line (Rudolph et al., 2019), by means of model-based techniques (Monclús et al., 2011), or by simple visual inspection of the trends of process variables such as the permeate flux and the TMP. However, monitoring fouling by means of these process variables might be cumbersome: they have been reported to exhibit strong variability (Philippe et al., 2013), which is in fact determined not only by membrane fouling, but also by the variability of process conditions, either natural or induced by deliberate control actions. On the other hand, the model-based approach has been successfully implemented to tackle similar problems, such as fouling in heat exchangers (Diaz-Bejarano et al., 2020), and found several applications to membranes as well (AlSawaftah et al., 2021).

Microfiltration and ultrafiltration can be generally described by the integral form of Darcy's equation (Meindersma et al., 1997; Whitaker, 1986), which relates the volume-flux of permeate and the TMP to the membrane resistance to flow (or to its reciprocal, the membrane permeability). In a way, membrane resistance represents the "health state" of a membrane and provides a measure of its fouling, as proved by several studies of membrane fouling focusing on resistance/permeability modeling and prediction (Dologlu et al., 2022; Geissler et al., 2005; Han et al., 2020, 2020; Huang et al., 2021; Kallioinen et al., 2006; Philippe et al., 2013; Ruiz-García et al., 2016). In the context of plant operation, the assessment of fouling through online monitoring of membrane resistance is of paramount importance to guarantee prompt processing

of products of upstream bioconversion, smooth downstream operation, and economical optimality of production in industrial biorefineries.

Estimation of membrane resistance (or permeability) by Darcy's equation is straightforward, provided that one has access to online measurements of permeate flux and TMP. However, even such simple demands might not be met in full-scale industrial processes, or even in pilot plants. Online measurements of permeate flux and TMP are typically available in plants employing a single membrane module (Chen et al., 2014; Han et al., 2020), but this is not usually the case when multiple modules are used, despite multi-module membrane separation being a common occurrence in industrial practice (Dologlu et al., 2022; Geissler et al., 2005; Kallioinen et al., 2006; Klimkiewicz et al., 2016; Ruiz-García et al., 2016). Given the limited availability of appropriate online data, only the average resistance/permeability of the ensemble of membranes is estimated, thus neglecting the actual resistances/permeabilities of single membrane modules. This clearly offers limited insight on the actual fouling state of the modules and hinders the identification of severe fouling events acting on single modules.

On the other hand, online measurements are not the only data source available in industrial processes. In fact, offline measurements are collected during process operation to monitor critical variables not available through online sensors, or that cannot be acquired automatically by cheap and/or reliable sensors (Kadlec et al., 2009). Therefore, available data are typically multi-rate, featuring online variables automatically acquired by the data acquisition system at high sampling rate and offline variables manually measured by operators at low sampling rate. While the time scale of acquisition of online variables is typically seconds or minutes, the time scale for offline variables is not as consistent and can vary from some hours to days or even weeks (Lin et al., 2009). Assuming, for instance, that available data for each membrane module operating in the plant consist of high sampling rate permeate flux measurements coming from the data acquisition system and low sampling rate TMP measurements coming from operator-read manometers installed on modules, single-module resistances can still be estimated at the TMP sampling rate (the lowest one). This solution might be unsatisfactory nonetheless, because the resolution of estimates could be too low to properly characterize relatively fast reversible fouling events, thus hindering punctual monitoring and prompt detection.

Despite the aforementioned limitations, many literature studies aim at modeling the evolution of membrane resistance by exploiting the information concealed in process data. Recent literature reviews (Bagheri et al., 2019; Velidandi et al., 2023) highlight that the most common approach is to consider only the average resistance in multi-module systems and to focus on either reversible or irreversible fouling, with limited attempts to resolve the two types (Chan et al., 2017; Huang et al., 2021; Klimkiewicz et al., 2016). Furthermore, strongly nonlinear models, such as neural networks, are used by default; these models require massive datasets (Rendall et al., 2019; Sun et al., 2021) to ensure robustness and to discern relevant phenomena (fouling) from common-cause process variability (in other words, to avoid overfitting).

However, such datasets are typically not available for large-scale processes (Rendall et al., 2019; Sun et al., 2021). On the other hand, simpler, linear modeling approaches are less demanding in terms of data amount, but are seldom used as they may require sophisticated measurements to achieve good performance, for example concentration of relevant compounds in the inlet stream acquired in specifically designed experimental campaigns (Philippe et al., 2013). Linear models can still achieve good performance in the prediction of TMP rather than resistance, as proved by Kaneko et al. (2013), but TMP may not be the most appropriate variable to monitor when the purpose is discriminating between reversible and irreversible fouling.

A different approach is to develop a soft sensor combining data and process knowledge in a hybrid modeling framework (Narayanan et al., 2023; Rajulapati et al., 2022; Sansana et al., 2021; Solle et al., 2017; von Stosch et al., 2014; Yang et al., 2020). Whereas the potential of soft sensors has been recognized to enhance sustainable process operation (Perera et al., 2023), their application to biorefineries and membrane separation processes at the industrial scale is still limited. In fact, few studies (Chan et al., 2017; Chew et al., 2017; Grisales Díaz et al., 2017; Hwang et al., 2009; Piron et al., 1997) considered the hybrid modeling approach, and most of them (Chew et al., 2017; Hwang et al., 2009; Piron et al., 1997) combined neural networks to predict the parameters of a knowledge-driven model (the cake filtration model) aimed at describing the evolution of resistance due to reversible fouling, thus neglecting irreversible fouling. Such studies considered a single-module pilot plant (Chew et al., 2017), or multi-module lab equipment (Piron et al., 1997) and pilot plants (Hwang et al., 2009), estimating only the average resistance in the latter cases. A different strategy was proposed by Grisales Díaz et al. (2017), where an artificial neural network was used to model two variables: the change rate (time derivative) of membrane permeability; a corrective term for TMP to account for possible osmotic pressure effects. Industrial data of a wastewater treatment plant employing a single membrane module were used. Chan et al. (2017) adopted yet another different approach, using the cake filtration model to estimate the energy requirement of single process runs and Gaussian process regression to model the prediction mismatch between runs, indirectly providing separate models for reversible and irreversible fouling. However, they achieved such results in a single, laboratory-scale membrane module operated under controlled fouling conditions.

In this study, we address the problem of characterizing both reversible and irreversible fouling in multi-module industrial biorefinery membrane separation systems by a hybrid modeling strategy that enables high-frequency estimation of the resistances of individual membrane modules. High sampling rate process data, together with low sampling rate TMP data, are first used to calibrate (and then use) a PLS model (Geladi et al., 1986; Wold et al., 2001) that estimates the TMPs of each membrane module at high frequency. Darcy's equation is then used to obtain high-frequency estimates of the resistances of each module. To test the proposed strategy, we use real data from two years of operation of the industrial scale biorefinery considered in this Thesis (Novamont S.p.A., 2016). The microfiltration section separates cells

from the bioconversion products and features seven interconnected membrane modules equipped with online sensors that measure the permeate fluxes; only offline manometers are available to measure the TMPs, thus requiring manual readings by operators. We show how monitoring individual resistances, even when done by simple visual inspection, can offer valuable insight on the reversible and irreversible fouling state of membranes. We also discuss the advantages of monitoring individual resistances, rather than TMPs and permeate fluxes, from the standpoints of data variability, effect of process changes, interaction between modules in multi-module systems, and dynamic evolution of fouling.

The remainder of this Chapter is organized as follows. The process investigated in this study and the available dataset are described in Section 4.2. Section 4.3 focuses on the mathematical models used to develop the soft sensor. Results are discussed in Section 4.4, and conclusions are drawn in Section 4.5.

4.2 Ultrafiltration process and data

In this Section, we introduce the first operation in the downstream chain: an ultrafiltration process carried out on porous membranes. We also discuss the current fouling monitoring strategy and the dataset available for the study presented in this Chapter.

4.2.1 Ultrafiltration process

The ultrafiltration unit processes the mixture from the upstream process and is a critical operation in the downstream train due to the high fouling potential of the feed (containing the biomass). A simplified process flow diagram of this operation is illustrated in Figure 4.2.

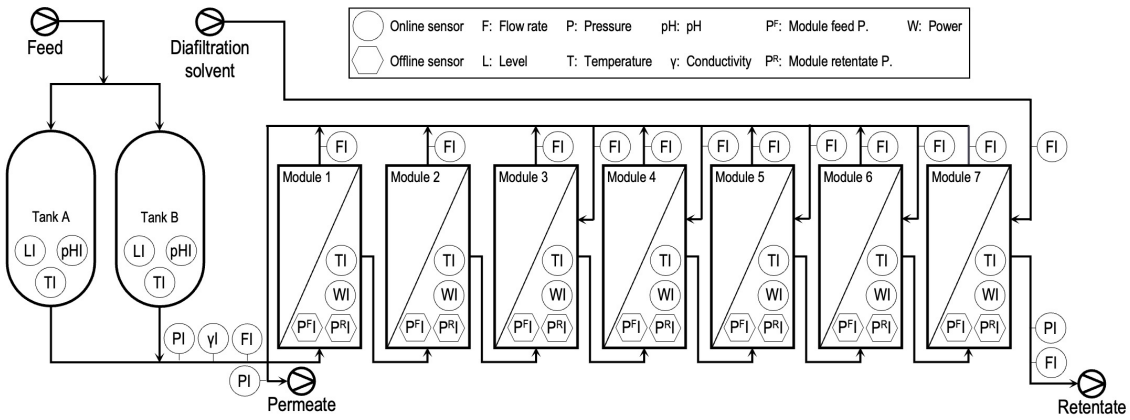


Figure 4.2. *Simplified scheme of the ultrafiltration operation in the downstream process.*

The sterilized broth is accumulated in two parallel feed tanks, which alternate in feeding an array of seven membrane modules filtering the broth to remove cells and high molecular weight compounds, yielding a clarified permeate stream containing the BDO. The retentate of a module

feeds the following one, while permeates of all modules are collected through a main manifold. The first two modules operate a simple concentration, while the remaining five employ a diafiltration strategy to maximize product recovery (Mulder, 1996). The feed flow rate is fixed, while flow rates of retentate and diafiltration solvent are adjusted based on preset ratios to the feed flow rate. The overall permeate flow rate (controlled variable) is kept constant by changing the overall TMP through feed pressure adjustment (by acting on the feed pump speed, manipulated variable).

The modules adopt the cross-flow configuration; therefore, the cross-flow velocities are adjusted by manipulating the speeds of the pumps incorporated into the modules (each speed can be manipulated independently) to counteract the effects of reversible fouling. Permeate flow rates of single modules are not individually controlled, but they can vary according to both membrane age (resistance) and applied TMP.

The process is interrupted, and cleaning is triggered when a preset volume of feed has been filtered or when the feed pressure exceeds a given threshold. Ultrafiltration is therefore run in semi-continuous mode, alternating operating and cleaning phases (Klimkiewicz et al., 2016; Philippe et al., 2013). The overall feed pressure and cross-flow velocity of one module on a selected timespan are reported as an example in Figure 4.3, where shaded intervals identify cleaning operations. Note that, for confidentiality reasons, all process variables will be reported as normalized values within the $[0, 1]$ interval in all figures throughout this Chapter.

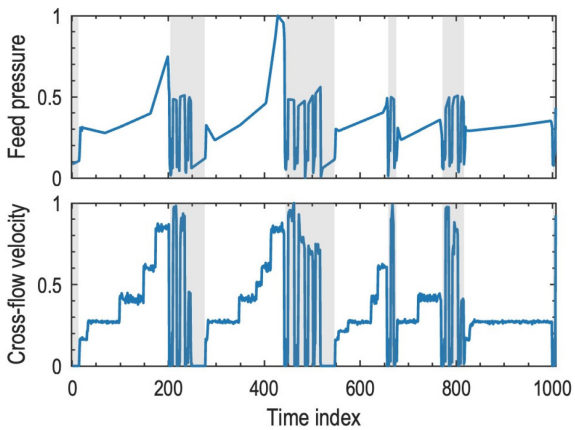


Figure 4.3. *Example of run-cleaning sequence in the ultrafiltration plant. Shaded intervals identify cleaning operations (large oscillations in the process variables are the result of the cleaning operations in these periods).*

4.2.2 Monitoring of membrane fouling in the ultrafiltration process

According to the current plant operation, the fouling state of membrane modules is monitored using offline measurements. Readings of manometers installed on each module can be used to compute the TMPs. Matching online measurements of permeate fluxes are then used to compute the flux-to-TMP ratios for each module, which are proportional to membrane permeabilities. An example is shown in Figure 4.4 for a period over which the membrane of

the relevant module was replaced (vertical solid black line), as can be clearly inferred by the trend of the flux-to-TMP ratio. Figure 4.4 also shows the effects of both reversible fouling (permeability increases after cleaning) and irreversible fouling (average permeability decreases across runs), though the former is harder to characterize properly due to the low frequency of readings. For example, a fouling event was detected on the third-to-last run shown in Figure 4.4 only after membrane permeability was largely degraded (see arrow in the figure), while an earlier detection would have helped operators to take action immediately.

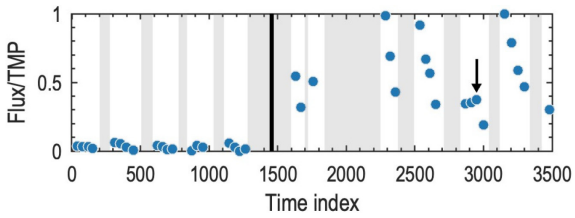


Figure 4.4. *Example of the effect of membrane replacement (vertical solid black line) on the flux-to-TMP ratio. Shaded intervals identify cleaning operations. The arrow indicates the last observation before a significant fouling event, which was not detected until the following observation due to the low frequency at which measurements are performed.*

4.2.3 Available dataset

Figure 4.2 illustrates all the online variables available through the data acquisition system. Level, temperature, and pH are measured for each feed tank. The feed manifold features flow rate, pressure, and conductivity sensors. Flow rate measurements are available for the retentate and diafiltration solvent manifolds, while pressure sensors are installed on the retentate and permeate manifolds. Each cross-flow membrane module is operated with controlled cross-flow velocity (inferred from the measured pump powers) and features sensors for temperature, permeate flow rate, and diafiltration solvent flow rate (where relevant). Online pressure sensors are not available on membrane modules, but manometers installed on the feed and retentate pipes allow for manual readings of pressure, which are made available as offline data.

The information conveyed by the 38 online variables and 14 offline variables is augmented by computing new variables. The overall volume-flow rate [$\mathrm{m^3\ s^{-1}}$] of permeate, $\dot{V}^{\mathrm{P}}$, is given by:

$$\dot{V}^{\mathrm{P}} = \sum_{l=1}^{7} \dot{V}_l^{\mathrm{P}} - \sum_{l=3}^{6} \dot{V}_l^{\mathrm{D}} \quad , \tag{4.1}$$

where $\dot{V}_l^{\mathrm{P}}$ and $\dot{V}_l^{\mathrm{D}}$ are permeate and diafiltration volume-flow rates [$\mathrm{m^3\ s^{-1}}$] of module l, respectively. The volume conversion ratio (VCR) of the multi-module system is computed as:

$$\mathrm{VCR} = \min\left\{\frac{\dot{V}^{\mathrm{F}}}{1\ \mathrm{m^3\ s^{-1}}}, \frac{\dot{V}^{\mathrm{F}}}{\dot{V}^{\mathrm{R}}}\right\} \quad , \tag{4.2}$$

$\dot{V}^{\mathrm{F}}$ and $\dot{V}^{\mathrm{R}}$ being the feed and permeate volume-flow rates [$\mathrm{m^3\ s^{-1}}$], respectively. A saturation is introduced by the minimum operator to limit the maximum VCR when $\dot{V}^{\mathrm{R}}$ approaches zero, typically during the startup and shutdown phases of each run. The TMP [Pa] of the multi-module system is defined as:

$$\Delta P = \frac{P^{\mathrm{F}} + P^{\mathrm{R}}}{2} - P^{\mathrm{P}} \quad , \tag{4.3}$$

where P^{F}, P^{R}, and P^{P} are the pressures [Pa] of the feed, retentate, and permeate manifolds, respectively. The multi-module TMP is hence defined as the difference between the average feed-side pressure (considering all modules) and the permeate-side pressure.

The variables available through relevant offline sensors are also illustrated in Figure 4.2. These variables are used to compute the TMPs of single membrane modules, the definition of which is similar to (4.3):

$$\Delta P_l = \frac{P_l^{\mathrm{F}} + P_l^{\mathrm{R}}}{2} - P^{\mathrm{P}} \quad , \tag{4.4}$$

but here P_l^{F} and P_l^{R} are offline readings of the feed and permeate pressures [Pa] of module l, respectively, while P^{P} is the corresponding online measurement of the pressure of the permeate manifold, assumed to be equal for all modules.

Data covering almost two years of operation are available for modeling. Data for some periods are missing due to changes in operational production. A total of 496 batches is found in the datasets. The number of observations of online variables spans between 21 and 260 per batch, with an average of 172, thus entailing a strong variability in the batch duration. On the other hand, offline variables are recorded between 1 and 6 times per batch, with an average of 4; most offline measurements are unevenly spaced and not aligned across runs. However, measurements of both online and offline variables are timestamped, thus observations can be matched. The available data are used to build a PLS model for online prediction of TMPs of all membrane modules using online variables only. This will be discussed in the next Section.

4.3 Hybrid soft sensor for membrane resistances

In this Section, the mathematical models used to build the hybrid soft sensor are introduced. The architecture of the overall model is presented as well.

4.3.1 Data-driven element: partial least-squares regression

The data-driven element used in the hybrid soft sensor is a PLS regression model, the rationale of which has been introduced in Section 2.2. For the following discussion, we assume that $\mathbf{Y}$ collects offline readings taken at low frequency, while $\mathbf{X}$ contains observations of process variables acquired online with timestamps matching the ones of offline observations in $\mathbf{Y}$. An in-depth discussion of the PLS element will be given in Section 4.4.1.

4.3.2 Knowledge-driven element: Darcy's equation

Permeation and filtration on dense and porous membranes are generally modeled through the transport theory. A thorough treatment of such a theory is out of the scope of this Chapter; interested readers are referred to notable literature resources (Baker, 2004; Mulder, 1996; Spiegler et al., 1966; Vilker et al., 1984).

Filtration through porous membranes can be described by the pore-flow model, represented by Darcy's equation (Darcy, 1856) in its integral form (Meindersma et al., 1997; Whitaker, 1986):

$$R = \frac{1}{\mu v}\Delta P \quad , \tag{4.5}$$

where R is the membrane resistance to flow $[\mathrm{m}^{-1}]$, μ is the dynamic viscosity [Pa s] of the permeate, v is the volume-flux $[\mathrm{m}^3\ \mathrm{s}^{-1}\ \mathrm{m}^{-2}]$ of permeate (permeate flow rate divided by the membrane surface), and ΔP is the TMP [Pa] (average pressure difference between the feed-side and the permeate-side of the membrane). Osmotic pressure is usually neglected for membrane filtration of bioconversion products because it is negligible for solutes with high molecular weight (Vilker et al., 1984; Wankat, 2009). Given the available online measurements, (4.5) allows to compute online the average resistance of the sequence of membrane modules in Figure 4.2 using the multi-module TMP computed as in (4.3). However, such "global" resistance is helpful to quantify the overall fouling state of the ensemble of modules but gives no insight on the actual states of single membranes. The resistance of each module can also be computed using offline measurements and applying Darcy's equation to each membrane module l computing the TMP from (4.4):

$$R_l = \frac{1}{\mu_l v_l}\Delta P_l, \quad l \in \{1, \dots, 7\} \quad . \tag{4.6}$$

The resistance computed in (4.6) represents the overall resistance of the membrane, including contributions from both reversible and irreversible fouling; it also has a well-defined physical meaning, as opposed to the "global" resistance computed by (4.5). However, since the frequency of offline measurements is low, the trend of the overall resistance does not allow to properly monitor reversible fouling. Therefore, we seek to obtain high-frequency estimates of the resistances of each membrane module in Figure 4.2.

4.3.3 Architecture of the hybrid soft sensor

A hybrid estimation approach is proposed, combining data-driven and knowledge-driven model components. Namely, offline observations of TMPs for each module, together with the corresponding observations of a subset of downsampled online and computed variables, are first used to calibrate a PLS model that estimates the TMP; subsequently, the model is used to obtain high-frequency estimates of TMPs based on online variables at their original sampling rate (details on the PLS model calibration and assessment are provided in Section 4.4.1). This corresponds to the data-driven component of the hybrid estimation approach. The knowledge-driven component is given by (4.6), which is used to estimate the resistances of all modules once the TMP estimates are available. The architecture of the soft sensor is graphically represented in Figure 4.5.

Viscosity measurements are not available online. Expert knowledge from plant operators was leveraged to assume a reasonable value for viscosity. Water-like behavior is postulated for permeate viscosities on all modules, and the effects of pressure and composition are deemed

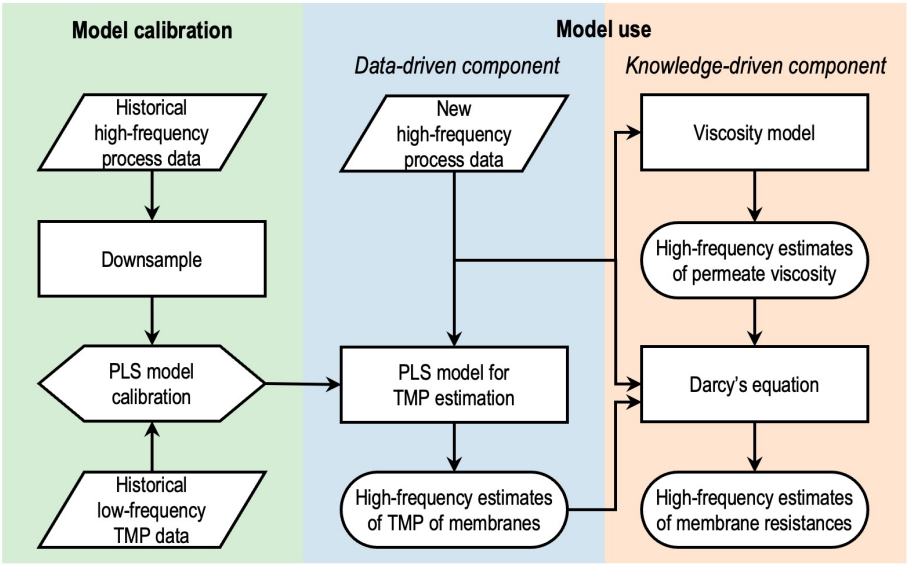


Figure 4.5. *Architecture of the proposed hybrid soft sensor. The middle column represents the data-driven component, while the column on the right represents the knowledge-driven component.*

negligible. The temperature effect on viscosity is modeled empirically (Perry et al., 2008):

$$\mu_l = \exp\left(-52.843 + \frac{3703.6}{T_l} + 5.866 \ln T_l - (5.879 \cdot 10^{-29}) T_l^{10}\right) \quad , \tag{4.7}$$

where the temperature is in K and the viscosity is obtained in Pa s.

Note that, even if the assumptions on viscosity could seem strong, the purpose of the soft sensor is not to yield extremely accurate estimates of the membrane resistance, but to accurately represent its trend to proper monitor the process. On the other hand, we are aware that an inaccurate estimation of the values of resistances could trigger membrane maintenance and replacement too early or too late, should the estimates be used to schedule such operations. Therefore, we recommend relying on the proposed model only for monitoring, while its use for maintenance scheduling is object of future research.

4.4 Results and discussion

Results of the study are presented in this Section. Section 4.4.1 presents the workflow for PLS model development and assessment, which are entirely achieved on MATLAB R2022a (The Mathworks, 2022a) with in-house-developed code. Sections 4.4.2 and 4.4.3 discuss the advantages of using resistances rather than fluxes and TMPs for process monitoring.

4.4.1 PLS model calibration and assessment

Direct prediction of membrane resistance by data-driven modeling usually requires strongly nonlinear models (Bagheri et al., 2019; Velidandi et al., 2023), while linear modeling may require to carry out ad-hoc experiments to acquire sophisticated measurements, for example concentration of foulants (Philippe et al., 2013). In fact, preliminary tests with a linear PLS

model to estimate resistances directly yielded very unsatisfactory results: as shown in Figure 4.7(b) residuals featured a clear trend not captured by the model, thus denoting the need for a nonlinear model; furthermore, significant autocorrelation was detected in the residuals, indicating that the process dynamics was not modeled, as reported in Figure 4.6(c). Therefore, as discussed in the previous Section, we tackle the problem by using a hybrid modeling approach: namely, we use a linear PLS model to provide estimates of TMPs from the available measurements; then, we couple this model to a simple nonlinear knowledge-driven model (Darcy's equation) to estimate the individual resistances from TMPs. A somewhat similar approach, yet in a different context, was used by Kaneko et al. (2013). The advantages of using a linear data-driven model are that, compared to a nonlinear data-driven model, model calibration is simplified, and model robustness is improved.

With respect to the PLS model, TMPs of membrane modules, computed by (4.4) from observations of offline variables at low frequency, are regarded as output variables, while input variables are the corresponding observations of a subset of online and computed variables, as reported in Table 4.1.

To build the PLS model, the online observations are downsampled to match the timestamps of online and offline observations; more sophisticated approaches to multi-rate modeling (Lin et al., 2009) were not needed for this study. Observations from all runs are stacked vertically to obtain the data matrices to be processed by the modeling algorithm. Stating the same in a

Table 4.1. *Input and output variables of the PLS model.*

ID	Variable	Symbol	Category
	Input variables		
X01	Feed flow rate	$\dot{V}^{\mathrm{F}}$	Measured
X02	Retentate flow rate	$\dot{V}^{\mathrm{R}}$	Measured
X03	Permeate flow rate	$\dot{V}^{\mathrm{P}}$	Computed by (4.1)
X04	Diafiltration solvent flow rate	$\dot{V}^{\mathrm{D}}$	Measured
X05	Overall TMP	ΔP	Computed by (4.3)
X06	Feed pressure	P^{F}	Measured
X07	Retentate pressure	P^{R}	Measured
X08	Permeate pressure	P^{P}	Measured
X09	VCR	VCR	Computed by (4.2)
X10 to X16	Permeate flow rates of modules 1 to 7	$\dot{V}_l^{\mathrm{P}}$, with $l \in \{1, \dots, 7\}$	Measured
X17 to X23	Temperatures of modules 1 to 7	T_l, with $l \in \{1, \dots, 7\}$	Measured
X24 to X30	Pump powers of modules 1 to 7	W_l, with $l \in \{1, \dots, 7\}$	Measured
	Output variables		
Y01 to Y07	TMP of modules 1 to 7	ΔP_l, with $l \in \{1, \dots, 7\}$	Computed by (4.4)

multivariate batch data analysis parlance, the structures of data matrices is the VWU one (Lee et al., 2004a). Cleaning periods are not included in the data matrices, as well as the startup and shutdown phases of each run, which usually feature excessive/unstructured variability and significant nonlinearities (Klimkiewicz et al., 2016). These preprocessing operations results in a $\mathbf{X}$ matrix in $\mathbb{R}^{1621} \times \mathbb{R}^{30}$ and a $\mathbf{Y}$ matrix in $\mathbb{R}^{1621} \times \mathbb{R}^{7}$.

The output variables feature a remarkable correlation (the minimum value is 0.9057), as can be inferred from the correlation matrix reported in Table 4.2. The correlation among outputs reflects the action of the control system, which adjusts the TMPs to compensate for the fouling state of each individual membrane. This guarantees the overall permeate flow rate to match the assigned set-point, while the permeate flow rates of single modules are free to vary according to the fouling state of each membrane. Since it is desirable to model such valuable information on the interaction among the modules, PLS is a natural choice due to its well-known ability to capture the correlations in input and output variables formulating LVs (Burnham et al., 1996).

Table 4.2. *Correlation matrix of the TMPs of the seven membrane modules (output variables of the PLS model).*

	ΔP_1	ΔP_2	ΔP_3	ΔP_4	ΔP_5	ΔP_6	ΔP_7
ΔP_1	1.0000	0.9485	0.9323	0. 9196	0.9141	0.9183	0.9057
ΔP_2	0.9485	1.0000	0.9570	0.9434	0.9424	0.9390	0.9417
ΔP_3	0.9323	0.9570	1.0000	0.9664	0.9692	0.9567	0.9417
ΔP_4	0.9196	0.9434	0.9664	1.0000	0.9678	0.9560	0.9432
ΔP_5	0.9141	0.9424	0.9692	0.9678	1.0000	0.9600	0.9561
ΔP_6	0.9183	0.9390	0.9567	0.9560	0.9600	1.0000	0.9561
ΔP_7	0.9057	0.9216	0.9417	0.9432	0.9561	0.9561	1.0000

The available data feature a dynamic component (due to the effects of fouling and the control system), which naturally calls for the use of dynamic PLS (Baffi et al., 2000; Dong et al., 2015; Ricker, 1988; Zhu, 2021). However, lagged matrices (on either data or scores) would then be required, which would set a constraint on the time step of observations in the prediction phase. Furthermore, lagged matrices require an even time step, while available data are sampled irregularly. Therefore, a static PLS model is developed. This allows one to calibrate the soft sensor with low sampling rate data, and to use it with high sampling rate data in the prediction phase. The choice of a static model for dynamic data is also backed up by two additional considerations. First, latent-variable models can capture dynamic information in data using additional LVs (Vanhatalo et al., 2016). Secondly, dynamic modeling is not necessarily needed to capture the input-output relation, as the dynamics of outputs could be solely induced by dynamics of inputs (Sun et al., 2021).

A preliminary PLS model is fitted on autoscaled data matrices by the SIMPLS algorithm (de Jong, 1993) selecting the number of LVs by repeated k-fold cross-validation with random partitioning of observations (Burman, 1989; Geisser, 1975) and one-standard-error-rule (Filzmoser et al., 2009; Hastie et al., 2009). The performance index is defined as the average of root-mean-squared errors of all 7 output variables, which results in 9 LVs. Data from nine runs are removed due to high values of the Q_X statistic for reconstruction of input data (Nomikos et al., 1995b), while data from three more runs are deemed as outliers and discarded due to high leverages (Berber et al., 2005; Rousseeuw et al., 1990). The preliminary PLS model is re-calibrated after removal of observations, which reduces the number or rows in the **X** and **Y** matrices to 1579; cross-validation results in 12 LVs.

Residuals of the new model feature remarkable dynamics, tested by the significance of coefficients of the autocorrelation function (ACF; Box et al., 2016) on 95% confidence limits computed by Bartlett's formula (Bartlett, 1946), as can be seen in Figure 4.6(a). Therefore, additional LVs are included in the model to remove as much residual autocorrelation as possible, improving the dynamics captured by the PLS model while preserving smoothness of the estimates. The fitted values and ACFs of the seven outputs are visually assessed to aid the tuning procedure, which results in a final PLS model with 20 LVs as best compromise. Figure 4.6(b) proves that most of the residual autocorrelation is removed. This approach was attempted also during the preliminary tests with the purely data-driven model for direct estimation of resistances, but significant dynamics was still left in the residuals even using all the available LVs. This further proves the value of the proposed hybrid modeling approach. As a matter of example, the ACF of residuals of one output of the PLS model for direct estimation of resistances with 20 LVs (the same as in the final PLS model for prediction of TMPs) is reported in Figure 4.6(c). For completeness, plots of residuals against fitted value (in calibration) of the PLS model for prediction of resistances and of the final PLS model for prediction of TMPs are reported in Figure 4.7. Residuals on the output variables used to compute the ACFs in Figure 4.6(b) and Figure 4.6(c) are reported in Figure 4.7(a) and Figure 4.7(b), respectively.

The generalization performance of the final PLS model are investigated by means of nested cross-validation (Varma et al., 2006) employing a repeated k-fold scheme with random partitioning of observations in both the inner and outer loops (Filzmoser et al., 2009). Such a tool is also used to make sure that the manual tuning of the number of LVs does not deteriorate the generalization performance. Average determination coefficients of the preliminary and final PLS model in calibration, validation, and testing are reported in Table 4.3. The final model shows a satisfactory generalization performance. Furthermore, determination coefficients in calibration and testing are similar, denoting that the model does not overfit even after manual tuning of the number of LVs. This fact serves as proof that the additional LVs are explaining dynamic effects, rather than unstructured variability found in the data, which offers assurance of the robustness of the model and reliability of its predictions.

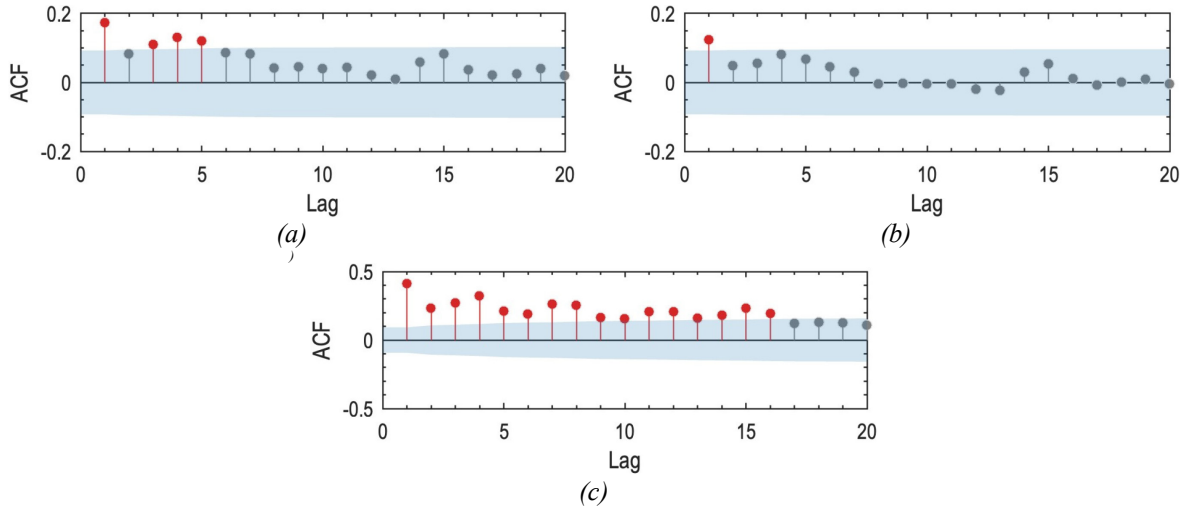


Figure 4.6. *Autocorrelation functions of residuals of an example output variable of the (a) preliminary PLS model with 12 LVs, (b) final PLS model with 20 LVs, and (c) initially investigated PLS model with 20 LVs for direct prediction of resistances. Significant coefficients are represented as red dots outside of the shaded envelope of the 95% confidence interval.*

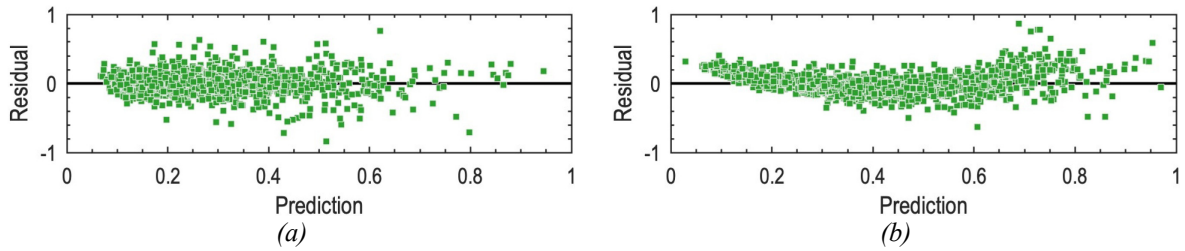


Figure 4.7. *Residuals of an example output variable of the (a) PLS model with 20 LVs for prediction of TMPs and (b) PLS model with 20 LVs for direct prediction of resistances.*

Table 4.3. *Average determination coefficients of the preliminary and final PLS models in calibration, validation, and testing estimated by nested cross-validation.*

Model	LVs	Calibration	Validation	Testing
Preliminary	12	0.9012	0.8979	0.8977
Final	20	0.9047	0.9005	0.9002

An example of prediction of the final PLS model using high-frequency online data is shown in Figure 4.8. The model successfully captures dynamic effects of both reversible and irreversible fouling. The effect of membrane replacements is excellently reconstructed, as can be seen in Figure 4.8: for instance, considering the second module, predictions "jump down" after the membrane replacement occurring around observation no. 1650, and this follows the trend of experimental data. Predictions tend to be unreliable at the very beginning of a run, which is especially clear considering the seventh module; however, they tend to realign to the actual TMPs in a relatively short time (which typically does not exceed 2 h and is comparable to the duration of the startup phase of a filtration batch). Prediction reliability can be assessed by means of the PLS model monitoring statistics introduced in Section 2.2.4, namely the T_X^2

statistic and the Q_X statistic (Nomikos et al., 1995b). Predictions are deemed as unreliable when Q_X is beyond its 95% control limit, while a T_X^2 beyond the limit denote that the process is drifting far from the average conditions. Control limits are computed according to the χ^2 distribution with matching moments approach (Nomikos et al., 1995a). Figure 4.9 reports such statistics in logarithmic form and allows one to clearly identify anomalous tails at the beginning of runs after observation no. 1950 in Figure 4.8, which are associated to statistics well beyond their control limits in Figure 4.9.

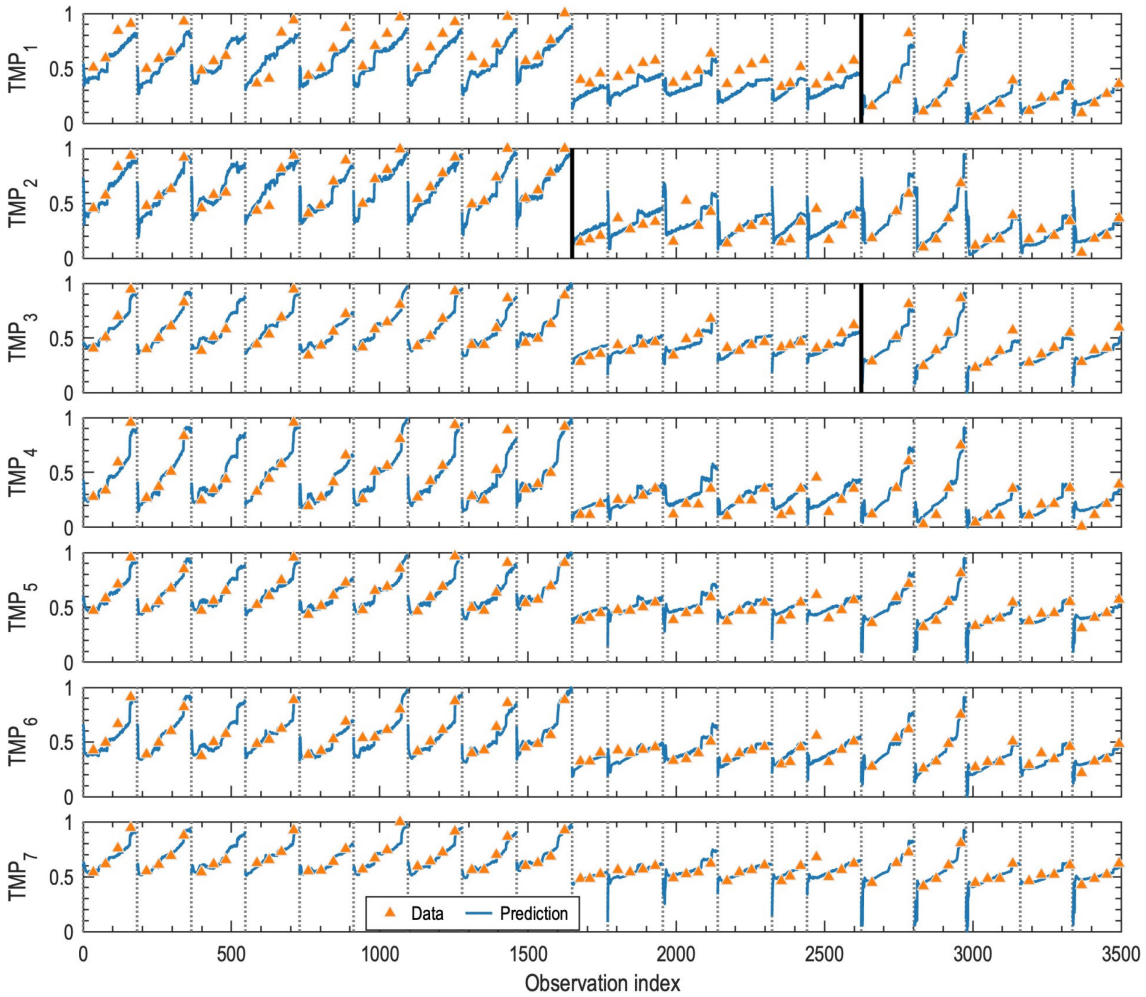


Figure 4.8. *Example of predictions of TMPs by the final PLS model. Low-frequency, offline measurements are represented as orange triangles, while high-frequency, online estimates are blue solid lines. Dotted lines delimit single process runs, while vertical solid black lines separate runs with a replacement of the membrane of the relevant module in between.*

4.4.2 Membrane resistances to monitor short-term fouling trends

The TMPs estimated by the PLS model are plugged in (4.6) with online measurements of permeate fluxes and temperatures (for the viscosity model) to estimate resistances of all membrane modules online. Resistances allow for a better monitoring and understanding of the filtration process as opposed to fluxes and TMPs. Some examples concerning the monitoring of short-term fouling are discussed in this Section.

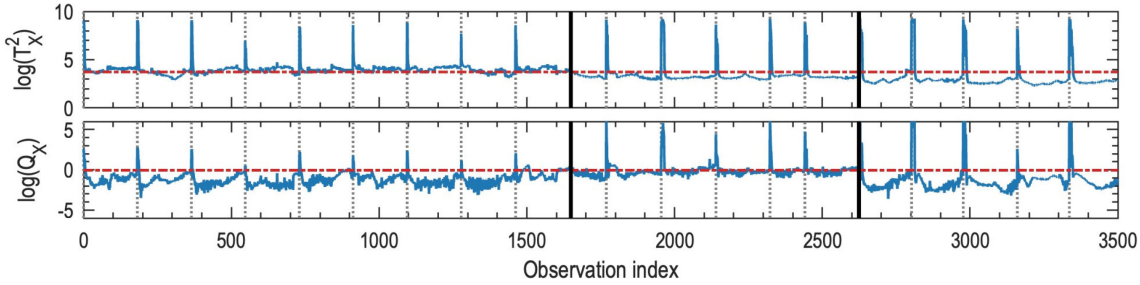


Figure 4.9. *Example of monitoring statistics to diagnose the reliability of predictions of the final PLS model; results refer to the predictions of Figure 4.8. 95% control limits are represented as red dash-dotted lines, vertical dotted lines delimit single process runs, and vertical solid black lines separate runs with a replacement of the membrane of any module in between.*

The increased interpretability is clear from Figure 4.10, reporting profiles of permeate flux, TMP, and resistance for one of the membrane modules in during two different example batches. In Figure 4.10(a) the estimated TMP features an increasing trend, whereas the permeate flux exhibits an erratic behavior with both fast and slow variations, which somewhat casts doubt on the interpretation of the TMP behavior. However, the estimated resistance features a well-defined, mostly monotonic trend, which allows one to unambiguously monitor the evolution of reversible fouling of the membrane under investigation along the batch. A significant fouling event is visible around batch time equal to 0.8, where the resistance steps up and then steadily increases thereafter; in fact, this specific batch had to be interrupted shortly after that event due to significant pressure build-up. Figure 4.10(b) highlights how membrane resistance can capture the occurrence of hydraulic cleaning within a batch. In this case, the material being filtered sedimented on the membrane surface in the initial part of the batch. Such deposition was removed due an increase in cross-flow velocity dictated by the control system shortly after batch time 0.3, which caused a decrease of the resistance. The increased flux however eventually enhanced membrane fouling, as can be argued from the rapid increase of the resistance after batch time 0.6.

For this particular example, the trends of membrane resistance and TMP are not too different, and one might think that both variables are equally effective to monitor short-term membrane fouling. However, that is not generally the case, and in fact monitoring resistances, rather than TMPs, offers significant advantages from the process understanding point of view. To appreciate this, recall that the plant layout (Figure 4.2) consists in several separation modules, but only the overall permeate flow rate is controlled, and this is achieved by manipulating the overall feed pressure. Therefore, a linear constraint acts on permeate fluxes of single modules, which are thus not independent and must compensate for each other. Such a strong correlation between fluxes makes it difficult to trace flux variations back to the fouling state of each single membrane, and this can impact also TMP trends. Furthermore, promptly identifying fouling events that act on single modules is cumbersome or impossible if done by visual inspection of the recorded data, due to the low frequency of offline readings. However, both issues are fixed

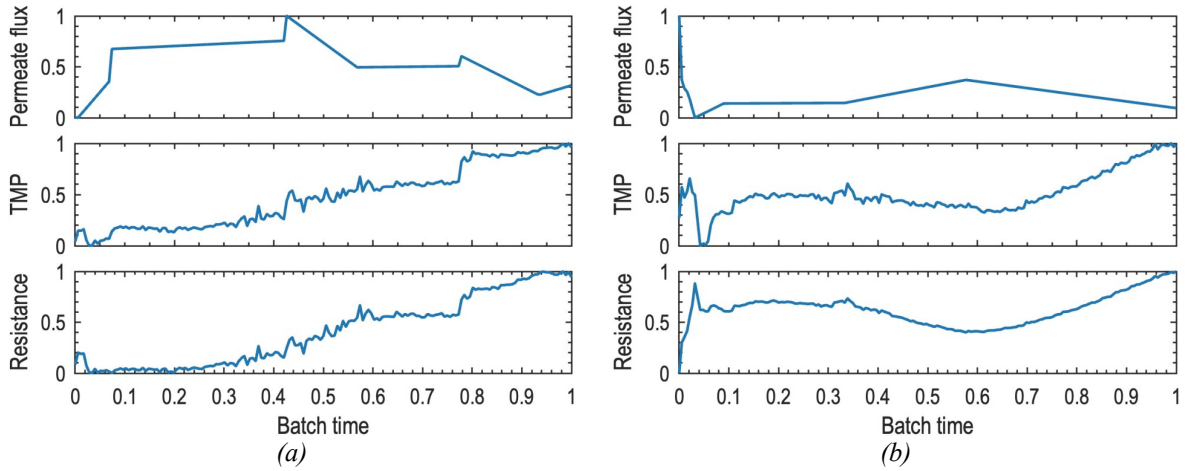


Figure 4.10. *Time profiles of fouling-related variables for one membrane module over two different batches to highlight the increased interpretability of (a) reversible fouling trend and (b) within-batch hydraulic cleaning phenomena due to the control system.*

if membrane resistance is monitored by the proposed approach, as elucidated in Figure 4.11 for two different batches. The effect of interdependence of permeates fluxes is clear from Figure 4.11(a), where fluxes of modules two to seven increase to make up for the decrease in flux of module one. However, TMPs of modules two to seven also increase, which makes it difficult to conjecture anything about the fouling state of membranes. On the other hand, resistances allow one to clearly understand that modules one and two are the ones mostly suffering from reversible fouling in this batch. Similar considerations can be done for the batch illustrated in Figure 4.11(b). Additionally, this figure enables one to appreciate that the onset of significant fouling events affecting single modules becomes clear if resistances (rather than TPMs or fluxes) are monitored. These events occurred in modules five and seven shortly after the batch start, and in module six around batch time equal 0.5, where the slope of the resistance changes.

4.4.3 Membrane resistances to monitor long-term fouling trends

The proposed model is helpful also in the analysis of long-term trends in fouling, which can be achieved by computing averages of the profiles of estimated resistances over batches. While this is possible also using low-frequency offline measurements, averaging the high-frequency estimates of resistances offers stronger reliability and increased robustness to outliers, thus allowing one to properly visualize and monitor long-term fouling trends for each single membrane module. We refer to such batch-averaged variables as features in the following.

The most prominent advantage of using estimated resistances for long-term fouling monitoring is the possibility to decouple membrane ageing effects from flux interdependencies across modules. As an example, consider Figure 4.12, which illustrates the trends of the permeate flux, TMP, and resistance features for one separation module across several consecutive batches. The end-of-life replacement of the module membrane occurs at batch no. 112 and is indicated by a solid black line in the figure. This causes the flux to step up in that module. However, due

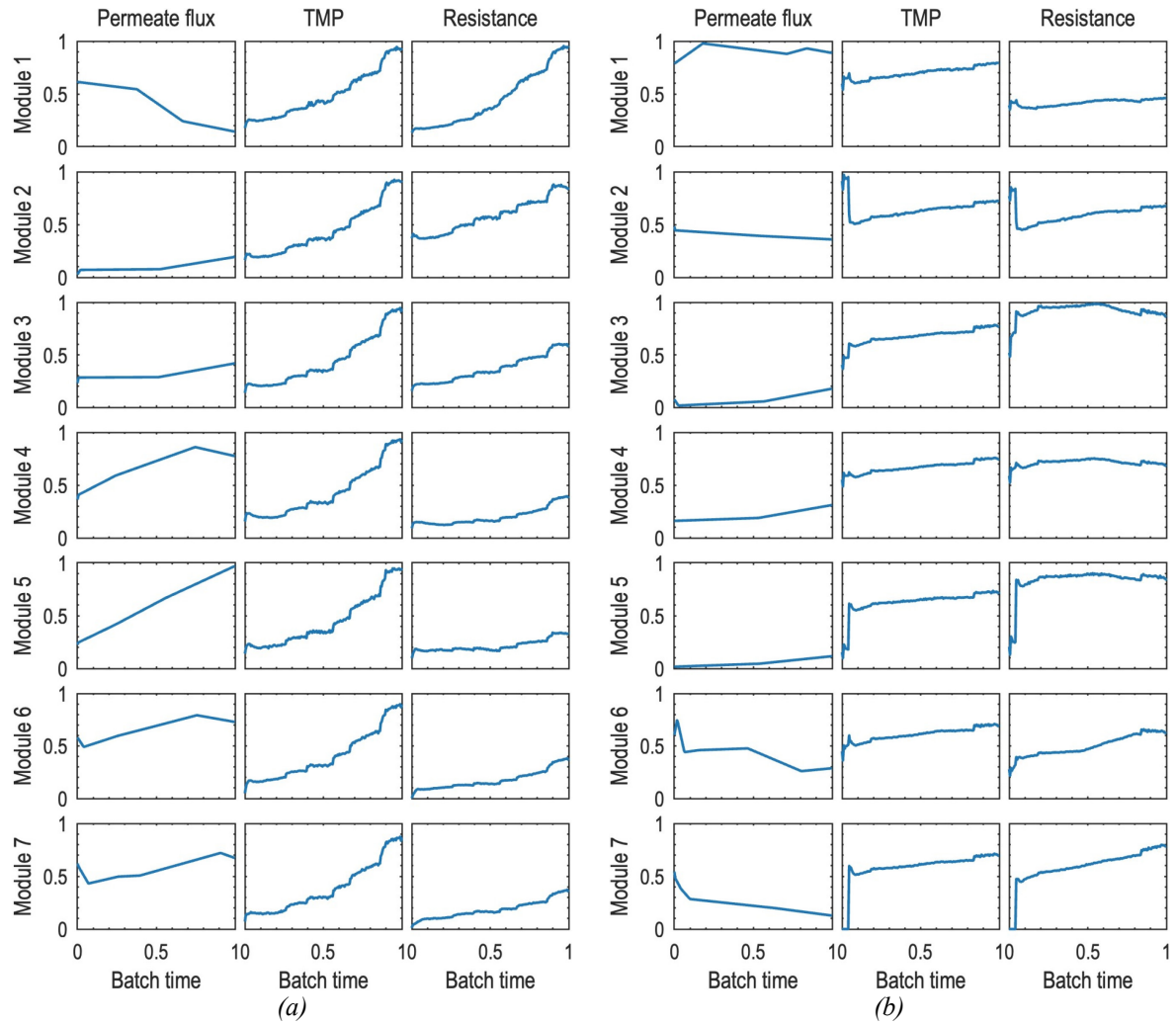


Figure 4.11. *Time profiles of fouling-related variables for all membrane modules over two different batches to highlight (a) the increased interpretability due to the monitoring of membrane resistances and (b) the identification of significant fouling events acting on a single module.*

to the interdependence of fluxes across modules, the other fluxes adjust accordingly (often stepping down). On the other hand, TMP values are so strongly affected by variability across batches that membrane replacement passes almost unnoticed. We conclude that analyzing fluxes or TMPs confounds the diagnosis of membrane health. Indeed, the true health state of the membrane is represented by its resistance, which clearly decreases right after replacement, thus facilitating monitoring. One additional advantage of monitoring the membrane resistance is that it does not change when the membranes in other modules are replaced (dashed lines in Figure 4.12); conversely, both TMP and flux are affected by the replacement.

The last, less apparent advantage of the proposed approach regards the decoupling of reversible and irreversible fouling, which can be achieved by combining the proposed model with knowledge of the process operation rationale. Features computed by averaging resistances over an entire batch mix up the effects of reversible and irreversible fouling into one single indicator. Intuitively, irreversible fouling is represented by the "baseline resistance" of a membrane at the

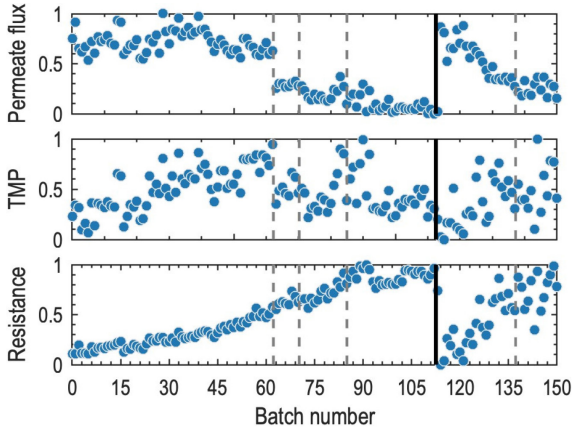


Figure 4.12. *Long-term trends of permeate flux, TMP, and resistance features of one membrane module across several consecutive batches. Each dot represents the average of profiles of the relevant variable on a batch. The vertical solid lines indicate replacements of the membrane of the relevant module, while vertical dashed lines are membrane replacements of other modules.*

beginning of a batch, and the within-batch increase of the resistance above the baseline is due to reversible fouling. This within-batch variation is compensated for by the process control systems, which adjusts the cross-flow velocity on discrete levels in response to reversible fouling, as shown in Figure 4.13: while the first two levels are always reached under normal operation, the following levels are enforced only when the feed pressure rises above preset thresholds. Reversible fouling is assumed to be under control (in the plant being considered) when the cross-flow velocity is set to the two/three lowest levels.

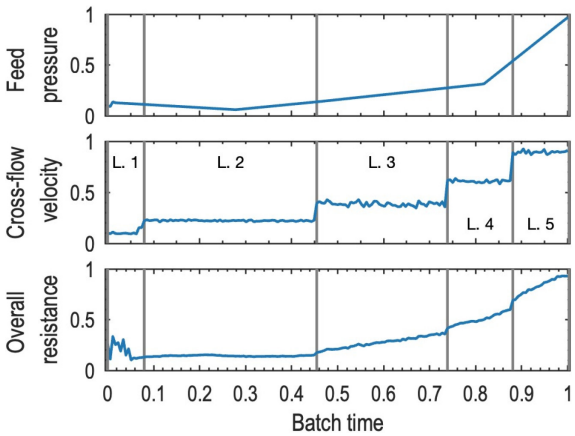


Figure 4.13. *Time profiles of fouling-related variables for one membrane module during one batch to illustrate the rationale of the variation of the cross-flow velocity on discrete levels (L.) to counteract the effects of reversible fouling. Time periods with constant cross-flow velocity are delimited by vertical lines.*

The outlined rationale can be exploited to decompose the overall resistance features into their contributions from irreversible and reversible fouling. The contribution due to irreversible fouling is computed as the average of the resistance profile where the cross-flow velocity is set to the three lowest levels; the contribution due to reversible fouling is computed as the

difference between the average of the resistance profile over the entire batch and the contribution due to irreversible fouling calculated as stated above. This decomposition is exemplified in Figure 4.14 for the membrane of one module. The overall resistance shows a clear long-term trend, which is associated to irreversible fouling and provides indirect indication of how the membrane state changes across a production campaign. However, batch-to-batch variability, due to the effect of within-batch reversible fouling, somewhat confounds the across-batch trend, particularly near membrane replacement (shortly after batch no. 290). The proposed resistance feature decomposition allows one to decouple the two effects: the resistance feature for irreversible fouling is affected by a much smaller variability, while the resistance feature for reversible fouling allows one to clearly identify batches that suffered from intense fouling, the profiles of which can therefore undergo additional, in-depth investigation.

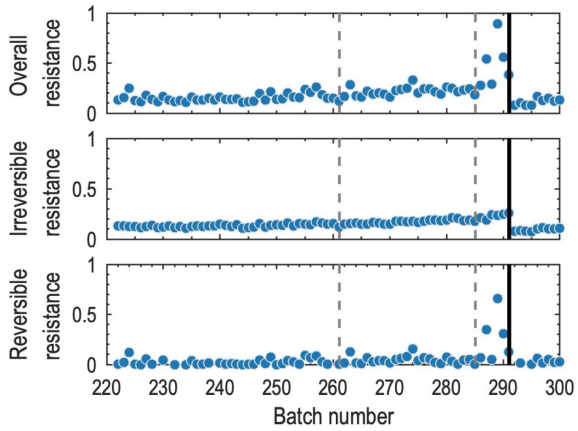


Figure 4.14. *Decomposition of overall resistance features (averages over complete batch profiles) of a module in their contributions form irreversible and reversible fouling. Vertical solid lines indicate replacements of the membrane of the relevant module, while vertical dashed lines are membrane replacements of other modules.*

4.5 Conclusions

In this Chapter, we developed a hybrid modeling strategy to estimate individual resistances of a multi-module membrane separation system of the industrial biorefinery considered in this Thesis. We combined a PLS regression model, for online estimation of the TMP of each membrane module, and Darcy's equation for modeling of membrane resistances. The proposed modeling strategy achieved excellent generalization performance using a linear data-driven modeling component, as opposed to the dominant literature approaches using nonlinear data-driven models that require massive datasets for calibration. To the best of the author's knowledge, this is the first time that such results were achieved on a complex industrial biorefinery process with limited data.

The main advantages of the proposed resistance-based monitoring approach for membrane fouling characterization are the reduced variability and increased interpretability of resistances with respect to permeate fluxes and TMPs. We illustrated examples of how resistances feature

clear and defined dynamics, allowing one to properly infer the fouling state of membranes and to promptly identify the onset of reversible fouling. We showed how the resistance of a module is independent of the resistances of other modules (therefore unaffected by replacements of other membranes in the system), a feature that fluxes lack when subject to linear constraints, such as a control system controlling the overall permeate flow rate. Finally, we discussed how to aggregate resistances as batch-averages to monitor the long-term evolution of fouling, proposing a method to decompose the overall membrane resistance in contributions from reversible and irreversible fouling by leveraging process knowledge. Results show that the contribution due to irreversible fouling features a monotonic trend and reduced batch-to-batch variability, while the contribution due to reversible fouling allows one to clearly identify batches that suffered from significant fouling issues, the profiles of which can be analyzed more in-depth for diagnostic purposes.

Chapter 5

Understanding membrane fouling by data-driven feature-oriented modeling[6]

A comprehensive analysis of membrane fouling in the ultrafiltration process of the industrial biorefinery considered in this Thesis is described in this Chapter. Feature-oriented modeling coupled with PCA is used to investigate the process settings most related to membrane fouling, aiming at identifying potential causes of the latter and devising strategies to mitigate it.

5.1 Introduction

In the previous Chapter, we presented a soft sensor based on a hybrid model for online estimation of membrane resistances in the ultrafiltration operation of the biorefinery considered in this Thesis. The model-based system serves as a tool for online monitoring of the fouling state of membranes, enhancing the biorefinery operation. Additional steps consist in fouling understanding and control, which are crucial for process improvement. Experimental methods for fouling understanding exist, but they often rely on invasive, sample-destructive procedures, such as membrane autopsy (Shi et al., 2014). Model-based methods can support fouling understanding as well and are the preferred asset to this end (AlSawaftah et al., 2021).

Model-based understanding of membrane fouling is mostly based on first-principles and mechanistic models centered on (4.5), the integral form of Darcy's equation (Meindersma et al., 1997; Whitaker, 1986), to estimate the membrane resistance to flow, which is then used as a direct measure of fouling. Therefore, models of the resistance change over time can be developed and interpreted to understand membrane fouling.

Reversible fouling and irreversible fouling (see Section 4.1 and Figure 4.1) are generally considered one at a time in the model-based approach. Concerning reversible fouling, the resistance in series model and Hermia's model can help diagnose the dominant mechanisms of reversible fouling (Juang et al., 2008; Vela et al., 2008; Wang et al., 2012). On the other hand, irreversible fouling is less understood and harder to model (Geissler et al., 2005), hence it is usually tackled by data-driven modeling (Dologlu et al., 2022; Han et al., 2020; Ruiz-García et al., 2016), often to purely predictive purposes.

[6] Part of the research discussed in this Chapter has been included in a manuscript in preparation (Arnese-Feffin et al., 2023d), to be submitted for publication as a journal paper. A preliminary version of this research work has been presented at an international conference (Arnese-Feffin et al., 2023c, 2023e).

While mechanistic models can provide detailed information about the dominant fouling mechanism and, potentially, its causes, they require proper experimental data to be calibrated (Bolton et al., 2006). Their applicability to industrial processes is hindered by the quality of data and by the limited span of operating conditions (Cuellar et al., 2020; Reis et al., 2018; Sun et al., 2021). However, the wealth of data provided by modern plants can still be capitalized: precious information on the nature of fouling can be extracted by data-driven modeling. This point has been demonstrated by several studies. Maere et al. (2012) applied PCA to investigate the fouling trend of a laboratory membrane bioreactor. A similar study was carried out by Kallioinen et al. (2006), who identified critical process variables affecting fouling in a pilot-scale pulp digester for paper production by means of PCA, DPLS, and parallel factor analysis. Naessens et al. (2017) exploited PCA to optimize the membrane cleaning schedule to minimize fouling in a pilot-scale desalination plant. An investigation on industrial-scale plant for wastewater treatment was carried out by Philippe et al. (2013): they used PLS regression to predict the fouling evolution in four parallel membrane units, and interpreted the models to identify critical process variables. Finally, Klimkiewicz et al. (2016) performed an analysis of fouling in an industrial ultrafiltration process treating fermentation broths, using multilevel simultaneous component analysis to investigate both reversible and irreversible fouling.

All the aforementioned studies dealt with a peculiar issue of membrane filtration: fouling causes the process to run semi-continuously (recall Figure 4.1), with alternating operating periods and cleaning phases, in such a way that an actual steady-state is basically never achieved. This causes membrane filtration data to assume the typical structure of the ones collected in batch processes, featuring three data dimensions: batches (runs), variables, and time. However, fouling causes a strong variability in the duration of batches (Arnese-Feffin et al., 2024; Klimkiewicz et al., 2016) and in the shape of profiles of process variables (Philippe et al., 2013), which additionally do not show the consistent “landmarks” typical of recipe-driven batch processes. These conditions undermine the applicability of traditional approaches to batch data analytics based on synchronization of batches (González Martínez et al., 2014a; Kassidas et al., 1998; Nomikos et al., 1994) followed by unfolding of the synchronized matrices (Camacho et al., 2009; Wold et al., 1987b, 1998), which can yield unsatisfactory results when strong variabilities in batch durations and shape of profiles of variables characterize the data at hand (Klimkiewicz et al., 2016).

In this Chapter, we adopt an alternative method to solve the lack-of-synchronization issue: feature-oriented modeling (Rendall et al., 2019; Yoon et al., 2001). We resort to knowledge-driven feature engineering (Wold et al., 2009) to design features capturing (and emphasizing) the information on fouling (Rendall et al., 2017a) starting from its observable effects. Knowledge-driven features also allow us to include process knowledge in the data analytics workflow (Severson et al., 2019). We focus on the ultrafiltration process of the industrial biorefinery considered in this Thesis (Novamont S.p.A., 2016). Using data from six months of

routine plant operation, we show how the analysis of features summarizing time profiles of process variables can shed light on membrane fouling in this complex industrial scenario and highlight some of its potential causes, paving the way for targeted experimental investigations. The reminder of the Chapter is arranged as follows. The process is described in Section 5.2; the fouling issues experiences by the plant and the data at hand for the investigation are described therein as well. Section 5.3 introduces the mathematical background of the model-based fouling investigation, including the proposed data analytics workflow to screen a large number of phenomena potentially unrelated to fouling. Results of the investigation are reported in Section 5.4, and conclusions are drawn in Section 5.5.

5.2 Ultrafiltration process and data

The ultrafiltration operation, already considered in Chapter 4, receives the solution containing the BDO from the array of upstream bioreactors and is aimed at separating cells and high molecular weight compounds. A simplified process flow diagram of this operation is illustrated in Figure 4.2. A general description of the process operation can be found in Section 4.2.1.

5.2.1 Observable effects of membrane fouling

Online readings (acquired through the online sensors reported in Figure 4.2) make possible to infer the overall fouling state of the ensemble of membranes. As an example, Figure 5.1 shows three process variables (overall permeate flow rate, feed pressure, and cross-flow velocity of a membrane module) in a sequence of run-cleaning cycles on a selected timespan, where shaded intervals identify cleaning phases. For confidentiality reasons, all variables will be reported as normalized values within the $[0, 1]$ interval in all figures throughout this Chapter.

While the feed flow rate is effectively kept constant, the feed pressure varies noticeably as a consequence of reversible fouling and, by (4.3) and (4.5), is directly proportional to membrane resistance. Therefore, in this Chapter, we focus on feed pressure as a measure of the fouling state of membranes. Figure 5.1 also highlights that, depending on the level of fouling of a membrane, increasing the cross-flow velocity could be enough to compensate for the effect of fouling, or it could not be able to do that. For example, Batch 2 and Batch 3 show increasing trends of the feed pressure due to fouling in their first parts, which are interrupted by the increase in cross-flow velocity, effectively compensating for the fouling buildup. On the other hand, increasing the cross-flow velocity is not sufficient to counteract reversible fouling in Batch 4 and Batch 5, where the pressure keeps increasing due to fouling buildup. Therefore, an investigation of the process settings most related to fouling could be highly beneficial to improve the operation of this process.

Note that this Chapter investigates the average fouling state of the overall membrane array, which can be inferred using data from online sensors only. Each module is also equipped with

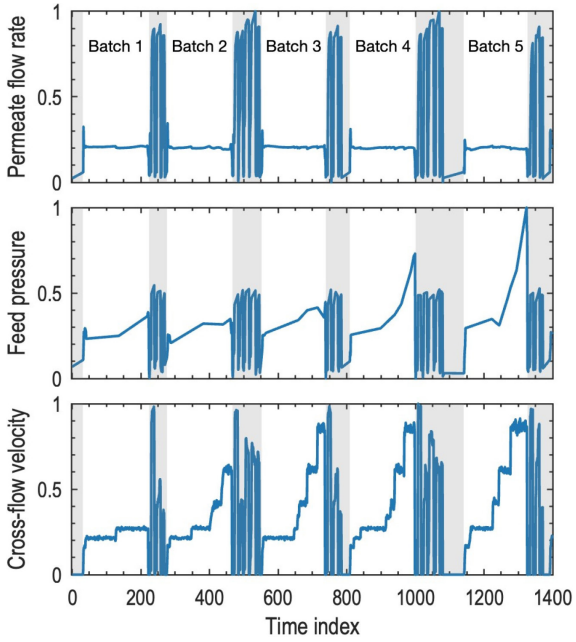


Figure 5.1. *Example of run-cleaning sequence in the ultrafiltration plant. Shaded intervals identify cleaning operations (large oscillations in the process variables are the result of the cleaning operations in these periods).*

offline manometers to measure the feed and retentate pressures (see Figure 4.2). While these variables would allow to compute the TMP of each module, hence to infer the fouling state of single membranes (although with low frequency), such a detailed analysis does not fall within the scope of this study. These additional data could be fruitfully exploited nonetheless, for example to calibrate a soft sensor for online resistance estimation, as done in Chapter 4.

While effects of short-term, reversible fouling can be clearly identified in Figure 5.1, the long-term effects of fouling can be visualized by means of permeate flow rates (or fluxes) of single modules. Figure 5.2 shows the average permeate flux of one of the membrane modules in a sequence of batches.

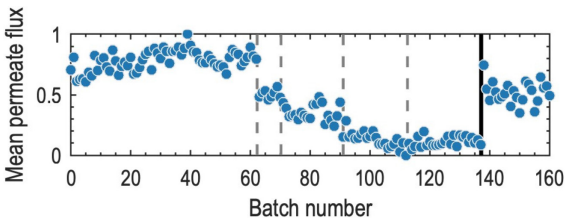


Figure 5.2. *Long-term trend of permeate flux of one membrane module across several consecutive batches. Each dot represents the average of the profile of the relevant variable on a batch. The vertical solid black lines indicate replacements of the membrane of the relevant module, while vertical dashed lines are membrane replacements of other modules.*

The decreasing trend of the permeate flux due to irreversible fouling is clearly visible in Figure 5.2. Furthermore, one can see that replacing the membrane of the relevant module (vertical solid line) leads to flux recovery. However, also replacements of membranes in other modules

(vertical dashed lines) can cause a flux variation. This is due to the overall permeate flow rate of the whole array of modules being controlled. Therefore, only the sum of single permeates is constrained by the overall material balance around the process, while contributions to it can vary and are determined by the fouling state of each membrane (similarly to a system of electric resistors in parallel), a point widely discussed in Section 4.4.2 and Section 4.4.3. This uncovers a strong coupling among membrane modules, which would make an analysis of fouling based on mechanist models extremely difficult. On the other hand, the wealth of data produced daily by the online sensors installed on the plant can be leveraged in a data-driven modeling scenario to improve process understanding and facilitate process conduction.

5.2.2 Available dataset

Data from six consecutive months of operation of the microfiltration process are available for modeling. After preliminary screening, 177 filtration batches are found in the dataset, with a high variability in the number of observations (time samples) per batch. Figure 5.3 shows that batch duration span between 65 and 238 observations per batch, with most batches counting around 195 observations. The application of profile synchronization approaches typically used in batch data analytics (González Martínez et al., 2014a; Kassidas et al., 1998; Nomikos et al., 1994) is hindered by the strong variabilities in batch duration and shape of profiles of variables across batches (see Figure 5.1), which do not show the typical landmarks of batch data (Klimkiewicz et al., 2016). On the other hand, this is the ideal environment for the application of feature-oriented modeling due to its ability to solve the lack-of-synchronization issue in a natural and direct way; no complex synchronization procedure is needed and simpler models with increased interpretability are generally obtained by feature-oriented modeling (Rendall et al., 2017a, 2019; Severson et al., 2019).

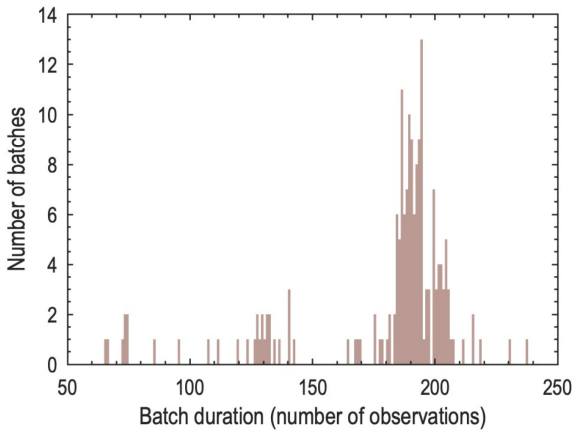


Figure 5.3. *Histogram of the duration of batches in the available dataset.*

Each batch in the dataset includes records of all variables measured online reported in Figure 4.2 and described in Section 4.2.3. Furthermore, some of the engineering variables described

in Sections 4.2.3 and Section 4.3.2 are computed to augment the dataset, due to the valuable information they provide:

- the overall permeate flow rate is computed as in (4.1);
- the VCR of the multi-module system is computed as in (4.2);
- the TMP of the multi-module system is defined as in (4.3);
- the average resistance of the ensemble of membrane modules is defined as in (4.5).

Profiles of additional variables are included in the dataset as well: the instantaneous energy consumption of the whole operation can be computed through the pump powers; characteristics of the feed, such as the biomass concentration, can be inferred from analyses performed in the upstream process and matching the production schedules of upstream and downstream; instantaneous slopes of the profiles of fouling-related variables (pressures, TMP, and resistance) can be obtained by numerical differentiation. A total of 62 variables is available in the final dataset. These data are processed for feature-oriented modeling, and the resulting features analyzed by PCA. The rationale of these methods is described in the next Section.

5.3 Data-driven investigation of membrane fouling

The mathematical models used for fouling investigation are introduced in this Section, together with the rationale of their application for the aim of the study. In particular, feature-oriented PCA modeling is used to investigate process settings related to fouling, aiming at identifying potential causes of this undesirable phenomenon and devising strategies to mitigate it. Features are generated from time profiles of variables leveraging the available process knowledge. We also propose a data analytics workflow to screen the large number of features originating from the several phenomena of interest for this study, which could or could not be related to fouling.

5.3.1 Feature-oriented principal component analysis

In this study, the data matrix, **X**, contains features computed from process data and PCA (see Section 2.1) is used for process understanding. This is achieved by interpretation of scores and loadings (Camacho et al., 2010; Kosanovich et al., 1996; Wold et al., 1987a). To this end, it is helpful to recall some of the properties of PCA that can be used for model interpretation.
Loadings describe the correlation among variables. Variables with high magnitude loadings on the same PC are correlated: if the loadings have the same sign, correlation is positive, while it is negative if the loadings have opposite signs. Furthermore, loadings describe how the PCs are formulated in terms of linear combinations of original variables, hence what change in original variables corresponds to a given change in the PCs. The latter change is described by the scores, which can be used to visualize the observation in **X** in a space of reduced dimensionality (the space of PCs). Observations with similar scores (similar in the space of PCs) are similar also in the space of original variables. Therefore, scores describe the correlation among observations.

PCA requires the process data to be arranged as a two-dimensional matrix. However, due to the semi-continuous nature of membrane filtration, data from these processes are commonly provided as a sequence of data matrices, as typical in data from batch processes. Since the batch duration changes across batches (recall Figure 5.3), the row dimension of the matrices in the sequence changes across batches as well. Such a sequence can be analyzed in a time-unresolved way by means of feature-oriented modeling (Rendall et al., 2019; Yoon et al., 2001).

The principles of feature-oriented modeling have been introduced in Section 2.6.2, where some possible methods to generate features from time profiles have been listed. While some of these methods enable for fast and automatic feature generation, they may not be entirely appropriate if the objective of the analysis is process understanding. To this end, knowledge-based features (Wold et al., 2009), also referred to as landmark features (Rendall et al., 2019), are a natural choice. This approach proved to significantly enhance the quality of models of complex phenomena, formulating the features to include physical knowledge on the system in the data analytics exercise (Severson et al., 2019).

Knowledge-based features are typically defined as transformations of the profiles of the original variables by simple mathematical operators selected based on the physical meaning of their results, so as to capture (and possibly emphasize) information on the phenomena of interest (namely, membrane fouling). Examples are averages, slopes, time integrals, minima, and maxima of variables in a batch. The rationale for feature design adopted in this study is described in the next Section.

5.3.2 Rationale of feature design

The knowledge-driven features used in this Chapter are based on simple mathematical operators. Process knowledge is leveraged before feature generation: the startup and shutdown phases of each filtration batch are removed from the data beforehand, as they are usually affected by excessive/unstructured variability and significant nonlinearities (Klimkiewicz et al., 2016). Features are therefore computed considering only the middle portion of profiles of variables in filtration batches, referred to as the steady phase herein.

Mean values of all process variables represent the most intuitive way to “compress” time profiles into scalars and are used as basic features. Time integrals of flow rates (total volumes) are included as well. The overall VCR, representing a whole batch and obtained replacing flow rates in (4.2) with overall feed and retentate volumes, is used as a feature as well. The run-time of each batch is included too. The duration of each sub-phase of the filtration (startup, steady phase, and shutdown) is considered as well. As the cross-flow velocity is varied on discrete levels by the control system to counteract reversible fouling (see Figure 5.1 and recall Figure 4.13), the duration of each one of the “level steps” are considered as additional features.

Averages, maxima, and average slopes of pressure-related variables are used as features to characterize reversible fouling. For example, the slope of the TMP vs. time curve is intuitively

related to the fouling rate (Monclús et al., 2011; Naessens et al., 2017), while the maximum TMP reached within a batch is related to the severity of reversible fouling. On the other hand, averages of permeate fluxes of single modules are considered as reliable indicators of the irreversible fouling state of membranes. The average, maximum, and average slope of the resistance of the multi-module system, computed as in (4.3), and the average energy consumption on a batch are considered representative of the overall fouling state of the system. Features characterizing the feed material and the upstream operations to produce it are included in the analysis as well. Examples are average feed properties (pH, conductivity, concentrations of biomass and salts, etc.), the bioreactor used to manufacture the feed being processed, and presence of contaminating microorganisms detected in a number of sampling points along the processing line. Finally, features denoting whether a given cleaning operation has been performed or not are considered.

Features related to the bioreactor used in the upstream, contaminations, and cleaning operations derive from categorical variables. Therefore, they are encoded using a dummy variables approach (Hastie et al., 2009): a categorial variables spanning on M discrete levels is represented by M binary variables, and the value of the m-th logical variable is 1 if the value of the associated categorical variable is on its m-th level, while it is 0 otherwise.

A total of 121 features is obtained for the analysis. Such a large number is due to the several phenomena that could be potentially related to fouling and to the variety of features used to encode them. However, not all the phenomena under investigation are necessarily related to membrane fouling. Therefore, some features are expected to be unrelated to the ones describing fouling. In the following Section, we propose a data analytics procedure for feature screening. Relevant features (namely, those discussed in Section 5.4) are reported in Table 5.1, along with their numerical identifiers, the category of physical phenomena they describe, and the numerical identifier of the models in which they are included (see Section 5.4).

5.3.3 Data analytics workflow

Given the complexities of the considered process and of membrane fouling taking place therein, several phenomena, which could or could not be related to fouling, need to be investigated. Therefore, a large number of features is obtained for PCA modeling, and many of them could be unrelated to fouling. This scenario is challenging for fouling investigation.

In this Section, we propose a two-stage data analytics workflow to identify the features that are most related to a given phenomenon in a large set of features that could or could not be related to the phenomenon of interest. The proposed workflow applied to the investigation of membrane fouling is schematically represented in Figure 5.4. The proposed approach is divided in two stages. The first stage (top box of Figure 5.4) concerns the development of a model to identify descriptors of membrane fouling and the batches particularly suffering from it. This is done calibrating a PCA model on a set of the fundamental features characterizing the process

Table 5.1. *Examples of features extracted from profiles of online variables, with the category of physical phenomena they describe and the PCA model in which they are used (refer to Section 5.4).*

			Used in model					
ID	**Feature**	**Category**	**0**	**1**	**2**	**3**	**4**	**5**
F005	VCR	Material balance	•	•	•	•	•	•
F008	Average feed flow rate	Material balance	•	•	•	•	•	•
F009	Average retentate flow rate	Material balance	•	•	•	•	•	•
F010	Average permeate flux	Material balance	•	•	•	•	•	•
F011	Average diafiltration flow rate	Material balance	•	•	•	•	•	•
F020	Average TMP	Reversible fouling	•	•	•	•	•	•
F021	Average slope of TMP	Reversible fouling	•	•	•	•	•	•
F022	Average feed pressure	Reversible fouling	•	•	•	•	•	•
F023	Maximum feed pressure	Reversible fouling	•	•	•	•	•	•
F024	Average slope of feed pressure	Reversible fouling	•	•	•	•	•	•
F029	Average energy consumption	Overall fouling	•	•	•	•	•	•
F030	Average resistance	Overall fouling	•	•	•	•	•	•
F031	Maximum resistance	Overall fouling	•	•	•	•	•	•
F032	Average slope of resistance	Overall fouling	•	•	•	•	•	•
F052	Processing delay time	Contaminations						•
F053	Average feed biomass concentration	Feed properties		•				
F054	Average feed conductivity	Feed properties		•				
F061	Contamination in upstream unit 1	Contaminations					•	
F062	Contamination in upstream unit 2	Contaminations					•	
F065	Cleaning operation 1	Cleaning		•				
F066	Cleaning operation 2	Cleaning		•				
F067	Cleaning operation 3	Cleaning		•				
F068	Cleaning operation 4	Cleaning		•				
F069 to F075	Average permeate fluxes of modules 1 to 7	Irreversible fouling			•			
F076 to F079	Average diafiltration flow rates of modules 3 to 6	Irreversible fouling			•			
F108 to F114	Average temperatures of modules 1 to 7	Temperature				•		

and membrane fouling. Significant PCs describing fouling-related features are identified in the model. In the study discussed herein, such model is described in Section 5.4.1.

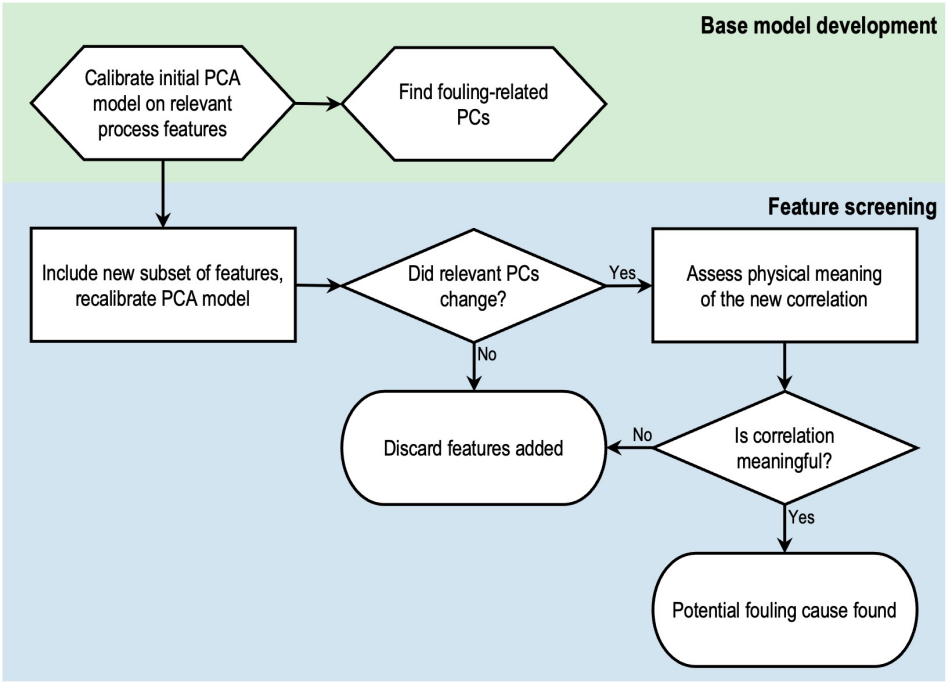


Figure 5.4. *Scheme of the proposed data analysis workflow for feature screening.*

The second stage (bottom box in Figure 5.4) is aimed at screening features unrelated to fouling and to identify the ones most related to the fouling-relevant features included in the PCA model developed in the previous stage. A new subset of features is included in the dataset and a new PCA model is calibrated. Loadings and scores on the significant PCs of the new model are visualized to spot any change with respect the PCs in the model from the first stage. If no relevant change in the PCs and scores is found (as in the case of the model in Figure 5.6), the new subset of features is deemed to be unrelated to fouling and discarded. If there are relevant changes (as in the case of Figure 5.7), the physical meaningfulness of the new correlation captured by the model is assessed: if it is not meaningful, the new correlation is deemed spurious, and the new subset of features is again discarded; if it is meaningful, then features in the new subset are deemed related to a potential cause of fouling, and engineering judgment is used to investigate such a cause. In the study discussed in this Chapter, the most relevant outcomes of the screening procedure will be outlined in Sections 5.4.2, 5.4.3, 5.4.4, and 5.4.5.

5.4 Results and discussion

The outcomes of the study carried out in this Chapter are discussed in this Section. Six models are described (refer to Table 5.1 for details on the features included in each model).

Model 0 The base-case PCA model developed with a subset of fundamental process features, including fouling-relevant features, described in Section 5.4.1.

Model 1 A PCA model including also features concerning cleaning operations and feed properties, described in Section 5.4.2.

Model 2 A PCA model including also features concerning irreversible fouling, described in Section 5.4.3.

Model 3 A PCA model including also features concerning temperature, described in Section 5.4.4.

Model 4 A PCA model including also features concerning upstream contaminations, described in Section 5.4.5.

Model 5 A PCA model including also features concerning processing delay time (related to contaminations as well), described in Section 5.4.5.

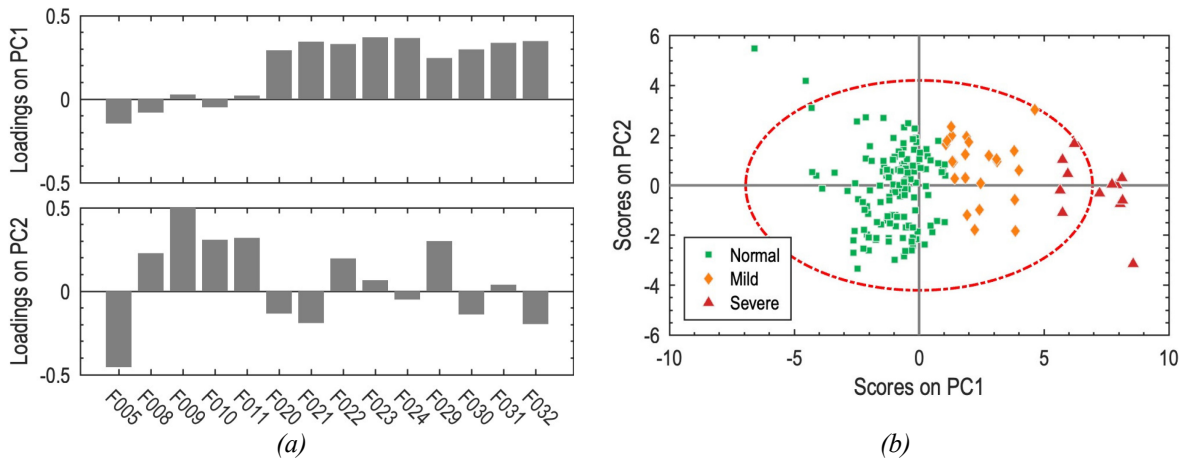
The data analytics workflow is implemented in MATLAB R2022a (The Mathworks, 2022a) with in-house-developed code.

5.4.1 Base-case PCA model

The data analytics workflow proposed in Section 5.3.3 begins with the calibration of a base-case PCA model (Model 0) including a set of fundamental features characterizing the process and membrane fouling. Such features are identified as:

- the overall VCR (feature F005);
- average flow rates/fluxes in the manifolds (features from F009 to F011);
- features related to the feed pressure (features from F022 to F024);
- features related to the TMP (features from F020 to F021);
- average energy consumption (feature F029)
- features related to the average resistance of the multi-module system (features from F030 to F032).

Two significant PCs are selected for Model 0, the loadings and scores of which are reported in Figure 5.5. The labels of features in Figure 5.5(a) correspond to the identifiers in Table 5.1. Figure 5.5(a) highlights that fouling-related features represent the major variability driver (first PC) of the process, as expected, while the second source of variability (second PC) is mostly



(a) *(b)*

Figure 5.5. *(a) Loadings and (b) scores of PCA Model 0 (the base-case), developed on the set of fundamental process features, including fouling-related features.*

related to the material balance of the system; the VCR (feature F005) and energy consumption (feature F029) impact the second PC as well. The model properly identifies batches that suffered from fouling issues, which are separated from the bulk of the runs in the score plot reported in Figure 5.5(b). Specifically, three groups are identified: normal batches (squares), batches suffering from mild fouling (diamonds), and batches suffering from severe fouling (triangles). The batches belonging to the three groups identified by Model 0 are represented with the same symbols used in Figure 5.5(b) in all of the following figures.

5.4.2 Effects of feed properties and cleaning operation

The feed of the ultrafiltration process (cells suspended in the solution containing the product) has a high fouling potential. This entails the need for frequent cleaning of membranes, which is achieved by means of four cleaning operations. Therefore, features concerning feed properties (features from F053 to F054) and cleaning operations (features from F065 to F069) are included in the dataset as a new subset in the feature screening phase of the proposed procedure, and a new PCA model is calibrated: Model 1. Figure 5.6 reports scores and loadings of the first two PCs of Model 1.

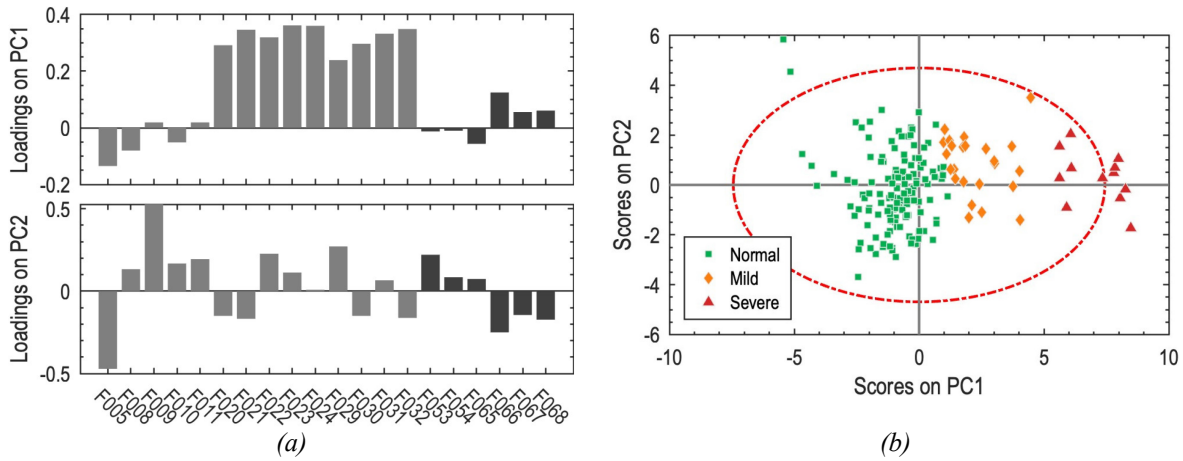


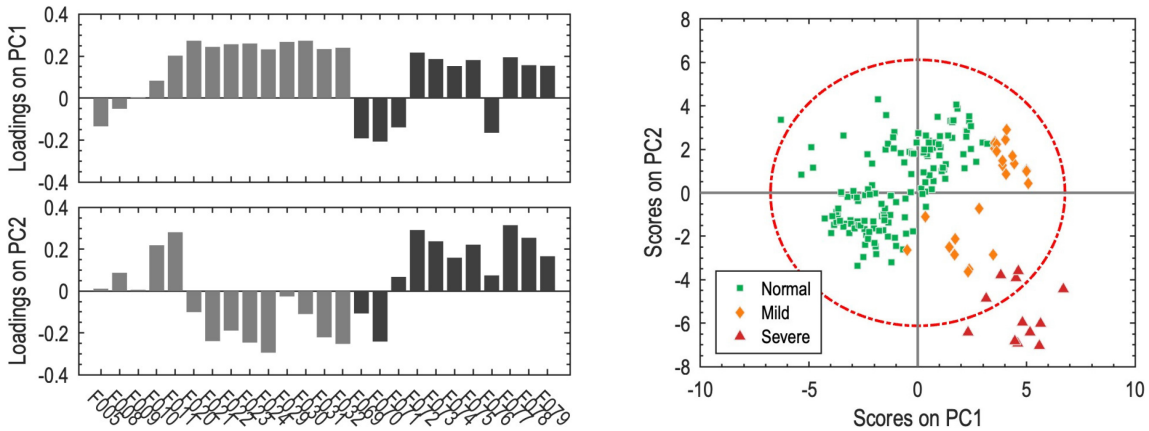
Figure 5.6. *(a) Loadings and (b) scores of the PCA Model 1, developed including features concerning the feed properties and cleaning operations. Loadings of the new subset of features are represented in darker color.*

The new subset of features is basically unmodeled by the first PC of Model 1, the loadings of which are shown in the top panel of Figure 5.6(a). This is confirmed by the score plot in Figure 5.6(b), as the grouping of observations is basically unchanged with respect to Figure 5.5(b). While this result might seem unexpected given the nature of the new features in Model 1, it provides valuable information nonetheless. The fact that features encoding cleaning operations (features from F065 to F069) do not show any correlation with the fouling-related ones (features F020 to F024, and from F029 to F031) confirms that the cleaning policies adopted in the plant are effective in removing reversible fouling. If cleaning operations were ineffective in contrasting reversible fouling, we would expect to find a negative correlation between the

sets of features representing the two phenomena (cleaning features are encoded as logical variables), denoting that batches not undergoing some cleaning operations have a higher fouling tendency compared to the batches where those operations are performed. Conversely, the lack of correlation between features representing cleaning operations and the fouling-related ones implies that the former do not explain any variability of the latter. Therefore, engineering judgement suggests that the cause of fouling is not related to the cleaning operations. On the other hand, it would still be reasonable to expect a high biomass concentration to imply severe fouling. However, the second PC shows a negative correlation between the VCR (feature F005) and the biomass concentration (feature F053), reflecting a control action performed by plant operators and consisting in lowering the VCR (by increasing the retentate flow rate) to compensate for high biomass concentration, thus effectively limiting the fouling rate. The effectiveness of this action is confirmed by the analysis of data discussed herein.

5.4.3 Interaction between reversible fouling and irreversible fouling

Model 0 includes features related to reversible fouling and to the overall fouling state of membranes. Features regarding irreversible fouling, namely average permeate fluxes (features from F069 to F075) and diafiltration flow rates of each module (features from F076 to F079), are included in Model 2 to investigate interactions between reversible fouling and irreversible fouling. Loadings and scores on the first two PCs of Model 2 are shown in Figure 5.7.



(a) *(b)*

Figure 5.7. *(a) Loadings and (b) scores of the PCA Model 2, developed including the average permeate fluxes and average diafiltration flow rates of single modules. Loadings of the new subset of features are represented in darker color.*

The new features show a significant correlation with features in the fundamental set, as can be seen in Figure 5.7(a), and cause a variation of the grouping pattern in the score plot, as reported in Figure 5.7(b). The loadings allow to conclude that reversible fouling is more intense when permeate fluxes of single modules are lower. This conclusion makes engineering sense, as permeate fluxes are proxies to the ages of membranes (thus their wear state, Figure 5.2), which is found to be the major factor related to reversible fouling issues. Since such fluxes are also

assumed to be reliable indicators of irreversible fouling, the newfound correlation points at the strong interaction between reversible fouling (manifested as high pressure reached during filtration) and irreversible fouling (in the form of a decrease of permeate flux batch after batch). Membranes operate in a normal way and reversible fouling is under control for most of their life, but their fouling propensity increases sharply after a given level of wear (irreversible fouling). This effect can be clearly visualized in Figure 5.8, reporting average permeate fluxes of three modules in the sequence of batches in the dataset.

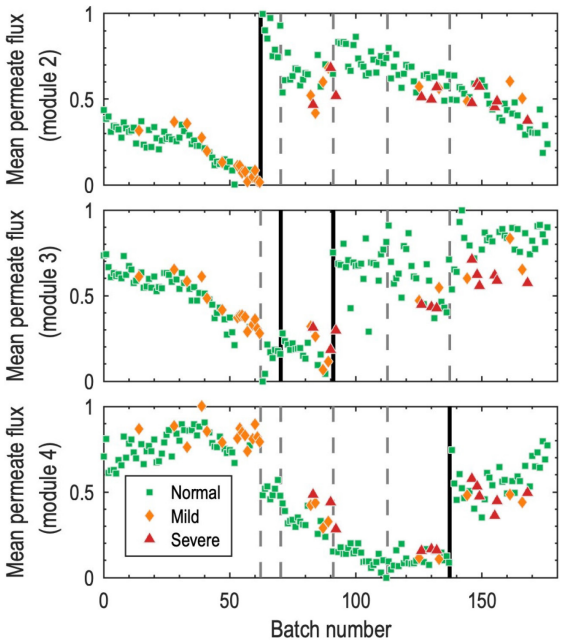


Figure 5.8. *Mean permeate fluxes of three selected membrane modules across several consecutive batches. Each point represents the average of the profile of the relevant variable on a batch. The vertical solid black lines indicate replacements of the membrane of the relevant module, while vertical dashed lines are membrane replacements of other modules.*

Batches suffering from reversible fouling are not evenly distributed in time but tend to group before membrane replacements. The analysis of values of fluxes (not reported for confidentiality reasons) highlights that a group appears anytime the flux of one of the modules falls below a given threshold. This information is extremely valuable for the operation of the plant, as it offers guidelines to improve the maintenance schedule of membrane units avoiding (or reducing) disruption due to fouling.

5.4.4 Effect of module temperature

The effect of temperature is investigated including the average temperatures of modules (features F108 to F114), yielding PCA Model 3, the scores and loadings of which are reported in Figure 5.9.

A clear correlation between fouling-related features and temperature features can be inferred from Figure 5.9(a), which is also reflected by the scores in Figure 5.9(b). The effect of

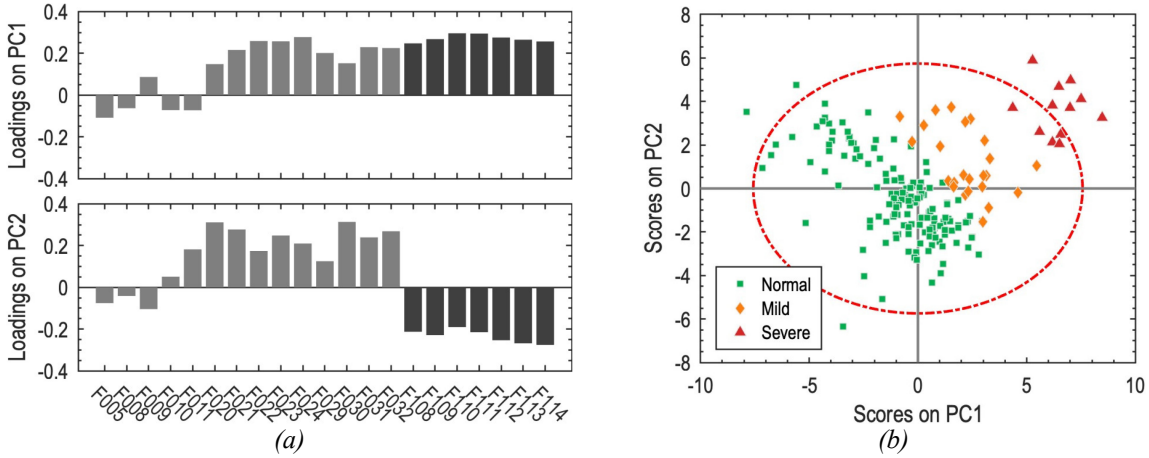


Figure 5.9. *(a) Loadings and (b) scores of the PCA Model 3, developed including the average temperatures of single modules. Loadings of the new subset of features are represented in darker color.*

temperature can be visualized in Figure 5.10(a), reporting the average temperature of a selected membrane module.

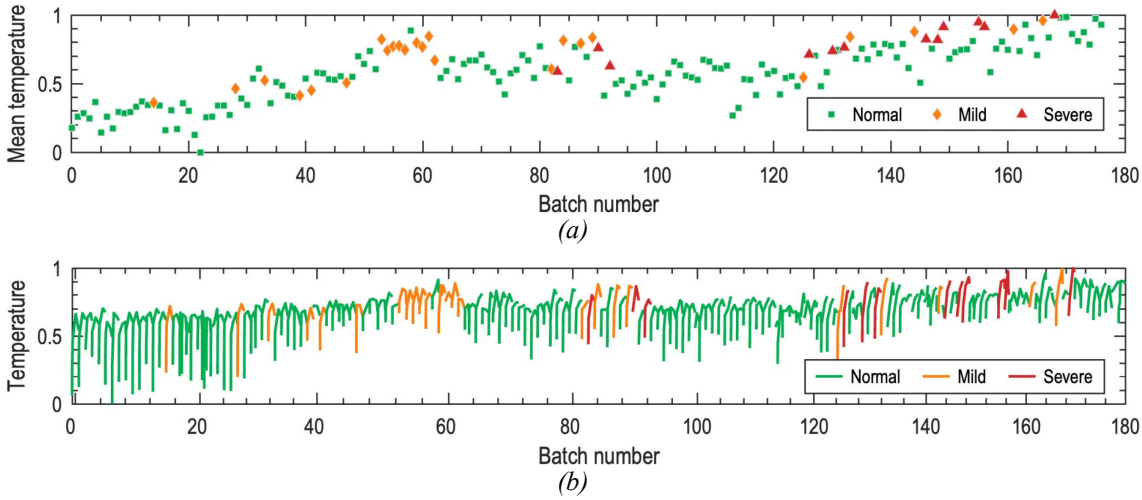


Figure 5.10. *(a) Mean temperatures of one membrane module across several consecutive batches. Each point represents the average of the profile of the relevant variable on a batch. (b) Steady-state phase of the raw profiles of temperature used to compute the features. Each profile is centered on the relevant batch number.*

The correlation between reversible fouling and temperature appears intuitively clear: the frequency and severity of fouling events increases when temperature increases. For comparison, Figure 5.10(b) reports the steady state phase of the raw profiles of temperature used to compute the features in Figure 5.10(a). While the trend of profiles is qualitatively the same as the one of features, interpretation of the former is more cumbersome due to the within-batch variability, which overlaps with the between-batch variability. Within-batch variability could even dominate the correlation structure of data, thus "distracting" the model from the phenomena of interest. The advantage of using feature-oriented modeling is apparent in this case, as they allow to "direct the attention of the model" to the phenomena of interest. More sophisticated

multiscale modeling approaches accounting for both within-batch and between-batch variabilities exist nonetheless (Bakshi, 1998; Klimkiewicz et al., 2016; Yoon et al., 2004).
However, a more in-depth analysis is in order concerning the effect of temperature. In fact, Figure 5.10(a) clearly highlights a seasonal variation of the temperature in the dataset at hand. This could mask the true effect of such a variable, causing the model to detect a spurious correlation which does not correspond to any causal relationship. Further doubts on the effect of temperature are casted by a deeper analysis of scores and loadings of the PCA model in Figure 5.9. One could note, in Figure 5.9(a), that the first PC models a positive correlation between fouling-related features and temperature (loadings of both groups are positive), while the second PC models a negative correlation (loadings of fouling-related features are positive, but loadings of temperate features are negative). This implies that the direction of fouling-related features in the score plot in Figure 5.9(b) is approximately aligned with the bisector of the first and third quadrants; on the other hand, the direction of temperature features approximately lies on the bisector of the second and fourth quadrants. Being the two directions nearly orthogonal, the two group of features appear to be independent rather than correlated.
Engineering judgement suggests that temperature should have an effect fouling on as it affects the viscosity of the fluids being processed and the characteristics of the possibly non-completely sterilized biomass in the feed. Furthermore, if biological fouling takes place (microorganisms attaching to and growing on membrane surfaces), temperature directly influences its growth kinetics and physical characteristics. Unclear effects of temperature were found also in other published studies concerning complex industrial scenarios similar to the one considered herein. For example, Philippe et al. (2013), highlighted that contradictory results were reported in the literature, concluding that there is no clear agreement regarding the effect of temperature on membrane fouling.
Given the outcomes of the model-based analysis described in this Chapter and the findings from the literature, we deem temperature as a potentially relevant factor for fouling. However, its effect is unclear, and we recommend the execution of a tailored experimental investigation.

5.4.5 Effects of upstream contaminations and processing delay time

An investigation of the effect of contaminating microorganisms is of interest for the present fouling analysis. Data regarding contaminations detected in a number of sampling points on the upstream processing line are available and used to design logical features identifying the origin of the contamination of the material being processed in the ultrafiltration unit (features F061 and F062). A PCA model including the upstream contamination-related features, Model 4 (loadings and scores not shown for brevity), is developed. No significant new correlation with fouling-related features is detected. This conclusion is supported by Figure 5.11, reporting one of the fouling-related features (the maximum feed pressure, feature F023) with vertical arrows marking ultrafiltration batches that processed a contaminated feed.

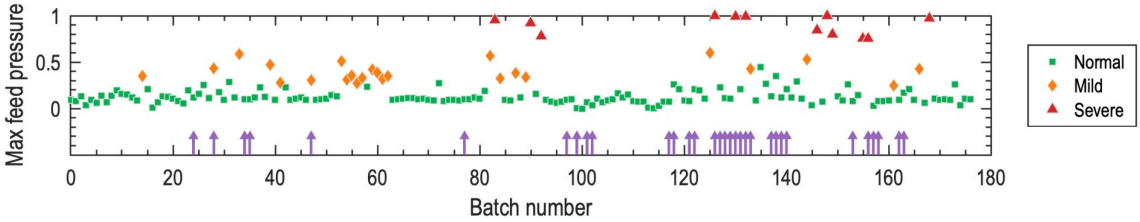


Figure 5.11. *Maximum feed pressure across several consecutive batches. Each point represents the maximum value of the profile of the relevant variable on a batch. The vertical arrows mark ultrafiltration batches that processed a feed found to be contaminated in the upstream.*

If upstream contaminations had any effect on fouling of ultrafiltration membranes, one would expect Figure 5.11 to show vertical arrows (contamination indicators) on most of the batches represented as orange diamonds or red triangles (the ones suffering from fouling issues). However, this behavior is not seen. In fact, batches suffering from fouling issues processed mostly non-contaminated feed, confirming that upstream contaminations do not significantly influence fouling of ultrafiltration membranes.

Another possible source of contaminations is the buffer tank between the array of upstream bioreactors and the sterilization unit. Depending on the production schedule of the plant, the fluid withdrawn from bioreactors could be withheld for some time in a buffer tank before sterilization, with the consequent risk of unpredictable behavior of the biomass, such as production of unwanted compounds or growth of contaminating microorganisms. A feature to account for this processing delay time (feature F052) is included in the PCA Model 5 (loadings and scores not shown for brevity) and is represented in Figure 5.12.

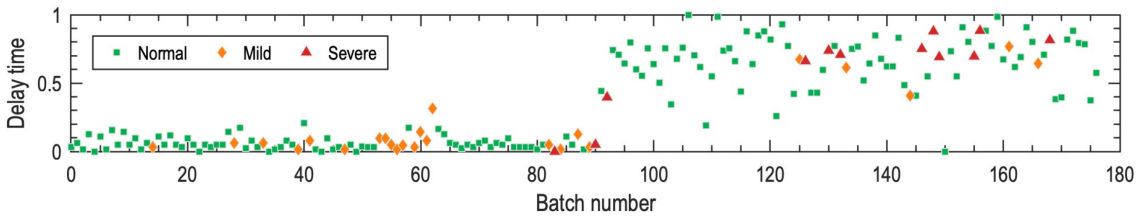


Figure 5.12. *Processing delay time across several consecutive batches.*

The processing delay time undergone a significant variation halfway through the timespan of data available for the analysis due to a change in the production schedule of the biorefinery, and a mild correlation with fouling-related features is detected by Model 5. However, one must note that the processing delay time (feature F052) also shows a remarkable correlation with features regarding temperature of modules (features F108 to F114), which highlights that the increase in hold-time in the buffer tank overlaps with the seasonal variation of the temperature. This phenomenon might lead the model to confound the two effects, therefore the effect of temperature should be made clear first, and the effect of the processing delay time should be assessed once this new information is available.

5.5 Conclusions

We carried out a comprehensive investigation of membrane fouling in the ultrafiltration operation of the industrial biorefinery considered in this Thesis. We resorted to interpretable data-driven models to develop knowledge on membrane fouling relying solely on data collected during process operation, and we proved how PCA can identify potential causes of fouling by analysis of data concerning the observable effects of this complex phenomenon, namely within-batch pressure rise (reversible fouling) and between-batch flux decline (irreversible fouling). A feature-oriented approach was adopted to cope with the strong variabilities in duration of filtration batches and shape of profiles of process variables induced by fouling: instead of using process data directly, numerical values characterizing each operating period were obtained to summarize time profiles into time-independent numerical features. We leveraged process knowledge to design features that could enhance the phenomenon under investigation, thus maximizing the dataset information content. A large number or features was obtained due to variety of process settings that could be potentially related to fouling, therefore we developed a systematic procedure for feature screening. Incorporation of process knowledge into the data analysis workflow proved essential to identify potential causes of fouling.

The model-based analysis allowed us to conclude that the cleaning policies and control actions currently adopted by plant operators can effectively manage reversible fouling up to a point where irreversible fouling of membranes is significant enough to neutralize such corrective actions, thus uncovering a strong interaction between the two fouling mechanisms in this plant. We also identified critical values of permeate fluxes of single modules (proxies to the irreversible fouling state of membranes) below which reversible fouling becomes hard to counteract by standard control/cleaning policies, providing precious guidelines for the improvement of the membrane maintenance schedule. Furthermore, we uncovered a potential relationship between temperature of membrane modules and severity of reversible fouling. However, the effect of temperature is unclear: contradictory conclusions were drawn by the analysis of data, therefore we recommended a tailored experimental investigation to shed light on this effect. Besides this, future work is directed to deeper investigations of the two fouling types. Detailed knowledge of the composition of the feed material would be beneficial to uncover the actual causes of reversible fouling by identification of the major foulants. On the other hand, modeling the dynamic evolution of irreversible fouling would enable us to set up proper predictive maintenance systems as to systematically optimize the cleaning and maintenance schedules of the plant.

Chapter 6

Regularized direct inversion to handle correlated quality variables[7]

A novel approach to algebraic inversion of PLS models is proposed in this Chapter. The popular DI approach to LVMI relies on the assumption that the variables describing the product quality are independent. However, product quality is often quantified by variables featuring varying degrees of correlation, which could cause singularities in DI. The method we propose in this Chapter can cope with this issue by design. The principle of the proposed approach is also leveraged to formulate an improved method to estimate the null space uncertainty.

6.1 Introduction

Ensuring consistent, on-target product quality and optimizing process conditions are daily challenges in the process industry. At the same time, the ever-growing availability of real-time measurements of several process variables, together with the potential of the Industry 4.0 paradigm, offers unique opportunities for boosting the performance of manufacturing processes to a new level (Reis et al., 2017, 2018, 2021b; Rendall et al., 2019; Venkatasubramanian, 2019). Interpretable data-driven models, such PCA (Wold et al., 1987a) and PLS regression (Geladi et al., 1986; Wold et al., 2001), allow translating data into information that can subsequently be capitalized into knowledge, thus shedding light also on complex processes for which principled knowledge may be lacking (for example biological processes). Multivariate statistical process monitoring systems (Kourti et al., 1995; Qin, 2003; Reis et al., 2017) or soft sensors for online estimation of product quality (Kadlec et al., 2009; Souza et al., 2016; Zhu et al., 2020) are just a few examples of successful applications of such latent-variable modeling techniques.

A PLS model relates input (predictor) and output (response) data from a process, and it also establishes models for the relevant data matrices. Therefore, if the matrix of inputs contains the available measurements on raw materials properties and process operating conditions (collectively referred to as process conditions in this Chapter), and the matrix of outputs is built on the available product quality measurements, the relevant PLS model encodes the relations between (a subset of) the process conditions and the product quality, while simultaneously

[7] Part of the research discussed in this Chapter has been published as a journal paper (Arnese-Feffin et al., 2022) and presented at an international conference (Arnese-Feffin et al., 2023a, 2023b).

modeling the process conditions and product quality themselves (Ferrer, 2020). This feature of PLS models makes them particularly attractive for addressing product design problems, namely for finding the combinations of raw materials and operating conditions that, on the one side, allow one to achieve an assigned product quality target, and, on the other side, are consistent with the past operation of the process (Jaeckle et al., 1998). From a modeling perspective, this task is known as LVMI (Jaeckle et al., 2000).

The rationale of LVMI can be stated as follows: set a target quality on the output variables, and run the model in "reverse mode", in such a way to calculate the values of the input variables that are most related to the assigned quality target according to the correlation pattern explained by the model. Since its original formulation, LVMI has been applied to several domains of industrial relevance, some of which have been listed in Section 1.2.4; Table 6.1 reports a more detailed list, with specific model inversion methodology used in the cited studies (the rationale behind each model inversion methodology has been briefly introduced in Section 2.5).

With respect to the formulation of the model inversion problem, Tomba et al. (2012a) proposed a general framework by systematizing contributions to LVMI coming from earlier studies

Table 6.1. *Some applications of latent-variable model inversion, and methodologies used to address model inversion. The methodologies are discussed in Section 2.5.*

Reference	Application	Model inversion methodology
Jaeckle et al. (2000)	Design of process conditions to achieve an assigned product quality in an industrial semi-batch emulsion polymerization process	Direct inversion
Yacoub et al. (2004)	Design of process conditions for an industrial insert-molding process	Inversion by optimization
Hwang et al. (2004)	Design of optimal environmental conditions for growing cells with desired levels of cellular functions in artificial organs engineering	Direct inversion
Flores-Cerrillo et al. (2004)	Control of simulated batch and industrial semi-batch polymerization processes	Direct inversion
García-Muñoz et al. (2005)	Product transfer between plants on a simulated low-density polyethylene polymerization reactor and scale-up of an industrial pulp digester	Direct inversion
Muteki et al. (2006)	Selection of optimal raw materials and blending ratios in an industrial polymer-blend production process	Inversion by optimization
García-Muñoz et al. (2006)	Design of reference profiles for an industrial pulp digester	Direct inversion; inversion by optimization
García-Muñoz et al. (2008)	Design of reference profiles for an industrial batch polymerization process	Inversion by optimization

Table 6.1 *(continued).*

Reference	Application	Model inversion methodology
Liu et al. (2011a)	Formulation of product and design of conditions of an industrial tablet manufacturing process	Inversion by optimization
Liu et al. (2011b)	Scale-up of a roller compaction process	Inversion by optimization
Tomba et al. (2013b)	Design of new quality profiles of pharmaceutical products from a lab-scale high shear wet granulation	Direct inversion; inversion by optimization
Tomba et al. (2014)	Product transfer between lab-scale units for the manufacturing of nanoparticles	Direct inversion; inversion by optimization
Facco et al. (2015)	Bracketing of design space of pharmaceutical processes demonstrated on three simulated case-studies	Direct inversion
Dal-Pastro et al. (2017)	Scale-up of an industrial wheat milling process with near-infrared measurements of product quality	Inversion by optimization
Bano et al. (2018b)	Identification of the design space in pharmaceutical processes demonstrated on three simulated case-studies	Direct inversion
Žuvela et al. (2018)	Preliminary screening of drug molecules given target molecular descriptors	Inversion by optimization
Zhao et al. (2019)	Design of operating parameters of simulated fed-batch penicillin production process	Direct inversion
Zhao et al. (2020)	Control of a simulated beer fermentation process	Direct inversion
Wang et al. (2020)	Identification of process conditions ensuring proper glycosylation in mammalian cell culture bioreactors	Direct inversion
Chu et al. (2021)	Transfer of a simulated cobalt oxalate synthesis process to a new plant	Inversion by optimization
Ruiz et al. (2021)	Achievement of non-defective products applied to real datasets of red wine and plastic pellets	Inversion by multi-objective optimization
Arce et al. (2021)	Design of optimal conditions for an analytical procedure to assess bisphenols while minimizing analysis time and sample volume	Inversion by multi-objective optimization

(Flores-Cerrillo et al., 2005; García-Muñoz et al., 2006, 2008; Yacoub et al., 2004) framing model inversion as an optimization problem. The proposed framework is applicable also when some quality variables have non-assigned targets and constraints have to be enforced in the

inversion (for instance, bounds of some of the input variables). The approach was further improved by Palací-López et al. (2019, 2020) with concern to the assignment of targets in terms of linear combinations of the output variables, as may occur, for example, when one needs to address economic target functions. An alternative formulation of LVMI based on a multi-objective optimization problem and exploiting the concept of Pareto optimality was proposed by Ruiz et al. (2018). Finally, extensions of LVMI to other, possibly nonlinear latent variables models have been proposed, such as nonlinear PLS (Yacoub et al., 2004), JYPLS (García-Muñoz et al., 2005), mixture-PLS (Muteki et al., 2006), total PLS (Zhao et al., 2019), kernel PLS (Zhu et al., 2021), and kernel JYPLS (Chu et al., 2021).

Most of the research on PLS model inversion focuses on the development of alternative approaches to formulate and solve the optimization problem arising in the inversion. However, no study has addressed the problem of improving the algebraic formulation of the model inversion problem, as originally proposed by Jaeckle et al. (1998). This formulation, also referred to as direct inversion (DI) in the literature (Tomba et al., 2012a), develops along a three-case workflow depending on the relative dimensions of the product quality space and of the process operating space, as discussed in Section 2.5.1. Whatever the relevant case, it assumes that the quality variables, upon which the response matrix is built, are independent (Jaeckle et al., 1998). However, in most practical cases the quality variables are correlated to some extent (Kourti et al., 1995; Wise et al., 1996). To circumvent this problem, Jaeckle et al. (1998, 2000) suggested two alternative approaches. The first one is to first build a PCA model on the entire set of quality data, and then use columns of the relevant score matrix corresponding to the significant PCs to build the response matrix. However, as noted by Jaeckle et al. (1998), a drawback of this approach is that some people may feel uncomfortable in using PCs instead of true variables to represent the product quality. The second approach is to build the response matrix by using only a subset of the quality variables. This approach fixes the above drawback; however, some of the information related to the quality variables space is lost when only a limited number of quality variables is used, which can make the PLS model not entirely representative of that space.

LVMI can yield infinite solutions in same cases (namely when the number of latent variables is greater than the number of quality attributes, see Section 2.5.1). In such cases, the subspace of solutions is referred to as null space (Jaeckle et al., 2000). In principle, any point along the null space should yield a product with the same quality attributes, a property that has been proved experimentally (Tomba et al., 2014). This gives additional degrees of freedom to process engineers to tune the solution to achieve some additional objective, say minimization of energy consumption, while still achieving the desired product quality (Jaeckle et al., 2000). However, the estimated null space suffers from uncertainty, an issue that must be accounted for in tuning the solution of LVMI. Several of approaches have been proposed to estimate the uncertainty of the null space (Bano et al., 2017, 2018a; Facco et al., 2015; Palací-López et al.,

2019; Tomba et al., 2012a). All these approaches are based on DI, therefore they inherit the assumption of independent quality attributes.

We propose an alternative formulation of the DI problem in order to improve the performance of algebraic model inversion and of null space uncertainty estimation in scenarios where the quality variables are correlated, exemplifying it in the case of PLS model inversion. As a side result, the proposed formulation simplifies the aforementioned tasks by resorting to a single-case workflow, with equivalent performance to the original case-based workflow developed for independent quality variables.

The remainder of this Chapter is organized as follows. In Section 6.2, we briefly recall the rationale of the mathematical methods used in this study, with a focus on the reasons why the existing algebraic formulation of the model inversion problem requires the model outputs to be independent, and we discuss the state of the art of PLS model inversion in the presence of correlated quality variables. We propose a new formulation of the LVMI problem, which can be used also when correlated outputs exist, in Section 6.3. We discuss advantages and limitations of the proposed approach therein and extend it to the estimation of the uncertainty of the null space. A comparison of the model inversion results for the existing and proposed formulations, with reference to two product design problems, is then presented: Section 6.4 refers to a simulated batch fermentation process, while Section 6.5 discusses a simulated fed-batch penicillin production process. Conclusions are drawn in Section 6.6.

6.2 PLS model inversion in the presence of correlated outputs

The mathematical methods relevant to this Chapter are PLS regression and LVMI in the form of PLS model inversion. The general form and rationale of PLS has been introduced in Section 2.2, while LVMI has been discussed in detail in Section 2.5, together with methods to estimate the null space uncertainty. In this Chapter, we assume that $\mathbf{X}$ and $\mathbf{Y}$ collect the measurements of process conditions and product quality variables, respectively.

6.2.1 Importance of the assumption of independent variables

Here, we provide a short discussion on the reasons why the original formulation of the model inversion problem (Jaeckle et al., 1998) is defined under the assumption of independent quality variables and according to a case-by-case workflow.

We first consider the case where the V_Y quality variables are indeed independent, thus $R_Y = \text{rank}(\mathbf{Y}) = V_Y$ in Figure 6.1(a). The reason why (2.75) can be used to calculate $\mathbf{t}_{\text{des}}$ if $A < V_Y$, but not if $A > V_Y$, is because it involves the right generalized inverse of $\widetilde{\mathbf{Q}}^{\mathrm{T}}$, which entails the inversion of $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$, a matrix in $\mathbb{R}^A \times \mathbb{R}^A$. Recalling that $\widetilde{\mathbf{Q}} \in \mathbb{R}^{V_Y} \times \mathbb{R}^A$, one can say that $\text{rank}\left(\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}\right) = \min\{A, V_Y\}$, and therefore $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$ becomes non-invertible if $A > V_Y$. Jaeckle et al. (1998) solved this issue by using (2.77) instead of (2.75) when $A > V_Y$, thus invoking a left

generalized inverse of $\widetilde{\mathbf{Q}}^{\mathrm{T}}$. This operation requires inverting $\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}}$, which is a full rank matrix in $\mathbb{R}^{V_Y} \times \mathbb{R}^{V_Y}$. In the special case of $A = V_Y$, $\widetilde{\mathbf{Q}}^{\mathrm{T}}$ is a square matrix, and it can be inverted directly as is (2.76).

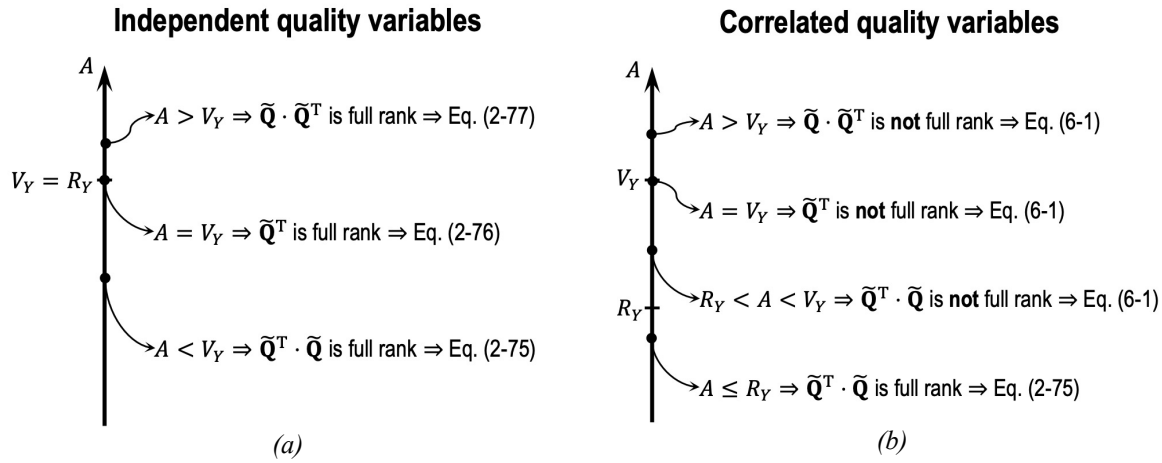


Figure 6.1. *Graphical interpretation of DI cases for different values of A with (a) V_Y independent quality variables, and (b) R_Y independent quality variables out of V_Y quality variables in total.*

Next, we consider the case where the quality variables are correlated, therefore $R_Y < V_Y$, meaning that the true dimension of the quality space is smaller than the number of quality variables, as shown in Figure 6.1(b). It follows that $\mathrm{rank}\left(\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}\right) = \min\{A, R_Y\}$. For the purpose of understanding the limitations of the DI formulation when correlated outputs exist, we consider a situation where one addresses the model inversion problem by naïvely disregarding the fact that the V_Y quality variables are correlated. Thus, one of the following cases would be encountered.

1. $A \leq R_Y$: the dimension of the latent-variable space is smaller than, or equal to, the dimension of the quality space. Matrix $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$ is full rank, and (2.75) can be used for DI.
2. $R_Y < A < V_Y$: the dimension of the latent-variable space falls between the dimension of the quality space and the number of quality variables. Matrix $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$ is not full rank because $A > R_Y$, and therefore it cannot be inverted as in (2.75).
3. $A = V_Y$: the latent-variable space has a greater dimension than the quality space, and this dimension is equal to the number of quality variables. Matrix $\widetilde{\mathbf{Q}}^{\mathrm{T}}$ is square but not full rank, because $V_Y > R_Y$, and cannot be inverted as in (2.76).
4. $A > V_Y$: the latent-variable space has a greater dimension than both the quality space and the number of quality variables. Matrix $\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}}$ is not full rank because $V_Y > R_Y$, and cannot be inverted as in (2.77).

In conclusion, when the quality variables are correlated, DI can still be performed by means of (2.75), but only if $A \leq R_Y$; however, if $A > R_Y$, none of equations (2.75), (2.76), and (2.77) can

be used, as all of them involve the inversion of matrices that are not full rank. That is why the assumption of independent quality variables is central to the workflow proposed by Jaeckle et al. (1998). We remark that this issue affects only DI and some of the cases of inversion by optimization, as it entails a matrix inversion. The inversion by multi-objective optimization approach mentioned in Section 2.5 does not suffer from it.

In the case $A > V_Y$, a null space exists and one can estimate its uncertainty, for example, by the analytical methods proposed by Facco et al. (2015) and Palací-López et al. (2019). This is possible in the case of independent quality variables only, as both methods involve the inversion of matrix $\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}}$ in (2.84) for Facco et al. (2015), and in (2.88) and (2.92) for Palací-López et al. (2019). Also other methods to estimate the null space uncertainty (Bano et al., 2017, 2018a; Tomba et al., 2012a) suffer from the same issue, being based on $\mathbf{t}_{\mathrm{des,p}}$ computed as per (2.77). However, the discussion is hereby limited to the methods by Facco et al. (2015) and Palací-López et al. (2019) by virtue of their simple analytical formulation.

6.2.2 Existing approaches to handle correlated quality variables

As discussed in Section 6.1, the existence of an issue in direct model inversion when the quality variables are correlated was early acknowledged by Jaeckle et al. (1998, 2000), who proposed two alternative ways to deal with it. The first approach consists in preliminarily building a PCA model on the full set of quality data. The columns of the score matrix of this model represent linear combinations of the true quality variables and are orthogonal by design. Therefore, they represent "artificial quality variables" that can be used to build the **Y** matrix, thus enabling the application of the case-based workflow discussed in Section 2.5.1. However, this has the disadvantage that people may not be comfortable in assigning targets to artificial quality variables rather than to true ones (Jaeckle et al., 1998).

As an alternative methodology, the same authors suggested to build the **Y** matrix by using a subset of the original quality variables, namely only those variables that can span the R_Y-dimensional quality space (Jaeckle et al., 1998). Although this is the preferred approach in the literature and can work effectively, it has some drawbacks. The first one is that, since only a subset of the quality variables is used, a subspace of the quality space is effectively ignored (unless the quality variables that are not included in **Y** are truly collinear[8] with the others), and therefore it is not described by the PLS model. Consequently, regardless of the model accuracy, the solution obtained by model inversion could not be able to ensure that the quality variables that have not been included in **Y** will be close enough to their targets[9]. The impact of this

[8] In this Thesis, the term "collinear" is used to denote 'a couple of variables that are linearly dependent', therefore that have unitary correlation coefficient. On the other hand, the term "correlated" means that 'the variables feature a given degree of correlation'.

[9] Note that this drawback also exists for the first alternative whenever the quality variables are not collinear and the number of principal components used in the PCA model is smaller than the number of quality variables.

drawback is not known in advance, because it depends on how strongly the quality variables are correlated, and on the quality variables that are included in **Y**. The second drawback of using only a subset of the quality variables is that some correlation between the variables that are included in **Y** always exists. Since process variability and measurement noise may partially mask this correlation, one may end up including in **Y** some quality variables that are correlated strongly enough to potentially give rise to an ill-conditioning issue in the matrix inversion calculations of (2.77) (García-Muñoz et al., 2006). Essentially, this issue is the same mentioned by Flores-Cerrillo et al. (2004), although in a slightly different context.

Some attempts have been done to formulate the model inversion problem in the presence of correlated outputs. Palací-López et al. (2019) and Wang et al. (2020) discussed the case-based workflow of DI considering rank R_Y in place of V_Y. García-Muñoz et al. (2006) extensively studied the concept of null space, including the case of correlated output variables; they proposed to handle the multiple solutions arising in the presence of such space by reformulating model inversion as an optimization problem in the space of LVs. They proved such approach to be effective and able to deal with ill-conditioning issues arising from the presence of correlation among quality variables. However, the solution of an optimization problem implies an increase of the computational cost. Zhao et al. (2019) extended the concept of LVMI to the total PLS modeling paradigm (Zhou et al., 2010), which is an extension of the PLS model based on the idea of "post-processing" the predictions of a PLS model to separate the output-relevant LVs (the ones actually relating the variance of the operating space to the variance of the quality space) from the output-irrelevant LVs (included in the PLS model to increase the explained variance of the operating space). Although not explicitly meant to address the issues related to correlated quality variables in LVMI, distinguishing LVs between output-relevant and output-irrelevant can be useful to account for correlation among output variables when performing DI of total PLS models, thus effectively solving the ill-conditioning induced by the presence of correlated quality variables. However, both model development and model inversion increase in complexity, with several hyperparameters to be tuned and processing operations required to the user in order to select an appropriate solution to the inversion problem in the presence of a null space. No algebraic approach to the inversion of a standard PLS model for cases where $A > R_Y$ has been proposed so far. A methodology to do this in a straightforward way is presented in the following Section.

6.3 Regularized direct inversion of PLS models

In this Section, we propose a novel framework that enables one to tackle output correlation by design in the formulation of the model inversion problem. Additionally, this will lead to a workflow that develops according to a single case, regardless of the relative values of A, V_Y, and R_Y.

6.3.1 Regularized direct inversion for collinear quality variables

We consider the case where only R_Y out of the V_Y quality variables are independent, while the remaining ones are collinear with the former ones; therefore, $R_Y < V_Y$, as in Figure 6.1(b). We assume that the output matrix $\mathbf{Y}$ is built by using all available quality measurements, and we propose an algebraic formulation of the PLS model inversion problem that can work effectively also in this condition. Extension to the case where the quality variables are correlated (but not collinear), or independent, is discussed in the next Section.

We first consider the case where $A > R_Y$, with $A < V_Y$. We have seen that numerical issues arise from the computation of the inverse matrix $\left(\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}\right)^{-1}$. We propose a LVMI method, named regularized direct inversion (RDI), that builds upon (2.75), but resorts to a matrix inversion algorithm exploiting regularization by means of SVD (Golub et al., 2013).

Letting $\mathbf{S} = \widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$ in (2.75), the first RDI equation becomes:

$$\mathbf{t}_{\mathrm{des}}^{\mathrm{T}} = \mathbf{y}_{\mathrm{des}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}} \cdot \mathbf{S}_{\mathrm{reg}}^{-1} \quad , \tag{6.1}$$

where $\mathbf{S}_{\mathrm{reg}}^{-1}$ is a regularized version of $\mathbf{S}^{-1} = \left(\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}\right)^{-1}$. SVD has been introduced in Section 2.1.1 in the context of PCA. However, in order to explain how $\mathbf{S}_{\mathrm{reg}}^{-1}$ is computed, it is worth recalling some facts about matrix inversion by SVD. Given that $\mathrm{rank}\left(\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}\right) = \min\{A, R_Y\}$, we have that $\mathrm{rank}(\mathbf{S}) = R_Y$, meaning that $\mathbf{S}$ is not full rank. SVD decomposes $\mathbf{S}$ as:

$$\mathbf{S} = \mathbf{N} \cdot \mathbf{\Sigma} \cdot \mathbf{O}^{\mathrm{T}} = [\mathbf{N}_1 \quad \mathbf{N}_2] \cdot \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \cdot [\mathbf{O}_1 \quad \mathbf{O}_2]^{\mathrm{T}} \quad . \tag{6.2}$$

The meanings of $\mathbf{N}$, $\mathbf{\Sigma}$, and $\mathbf{O}$ have already been introduced in Section 2.1.1. However, belonging such matrices to the SVD decomposition of the square matrix $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$, they are all square matrices in $\mathbb{R}^A \times \mathbb{R}^A$. As $A > R_Y$, only the first R_Y singular values of $\mathbf{S}$ are non-null and are collected on the main diagonal of $\mathbf{\Sigma}_1 \in \mathbb{R}^{R_Y} \times \mathbb{R}^{R_Y}$. Consequently, the remaining $A - R_Y$ singular values are null because of collinearity among responses, and matrix $\mathbf{\Sigma}_2 \in \mathbb{R}^{A-R_Y} \times \mathbb{R}^{A-R_Y}$ is actually the null matrix $\mathbf{0}$. Coherently, $\mathbf{N}$ is expressed as a block matrix made by $\mathbf{N}_1 \in \mathbb{R}^A \times \mathbb{R}^{R_Y}$ and $\mathbf{N}_2 \in \mathbb{R}^A \times \mathbb{R}^{A-R_Y}$, while $\mathbf{O}$ contains $\mathbf{O}_1 \in \mathbb{R}^A \times \mathbb{R}^{R_Y}$ and $\mathbf{O}_2 \in \mathbb{R}^A \times \mathbb{R}^{A-R_Y}$.

If $\mathbf{S}$ were a generic full-rank matrix, its inverse could be computed from (6.2) according to:

$$\mathbf{S}^{-1} = \mathbf{O} \cdot \mathbf{\Sigma}^{-1} \cdot \mathbf{N}^{\mathrm{T}} \quad , \tag{6.3}$$

However, since $\mathbf{S}$ is not full rank, $\mathbf{\Sigma}^{-1}$ cannot be computed as diagonal elements of $\mathbf{\Sigma}_2$ are zero. The inversion of matrix $\mathbf{S}$ with rank R_Y can be regularized by neglecting the last $A - R_Y$ singular values and singular vectors, hence by neglecting all matrices with subscript 2 in (6.2), which yields the second RDI equation:

$$\mathbf{S}_{\mathrm{reg}}^{-1} = \mathbf{O}_1 \cdot \mathbf{\Sigma}_1^{-1} \cdot \mathbf{N}_1^{\mathrm{T}} \quad . \tag{6.4}$$

So far, we presented RDI with reference to the case $A > R_Y$ with $A < V_Y$, which is only one of the cases illustrated in Figure 6.1(b). However, (6.1) and (6.4) hold true for all cases in which $A > R_Y$, because $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$ is an $\mathbb{R}^A \times \mathbb{R}^A$ matrix with rank R_Y regardless of the relationship between A and V_Y. On the other hand, if $A \leq R_Y$, then $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$ is a full-rank $\mathbb{R}^A \times \mathbb{R}^A$ matrix;

therefore, it features no null singular value; this implies that $\mathbf{N} = \mathbf{N}_1$, $\mathbf{\Sigma} = \mathbf{\Sigma}_1$, and $\mathbf{O} = \mathbf{O}_1$, from which we derive that $\mathbf{S}_{\text{reg}}^{-1} = \mathbf{S}^{-1} = \left(\widetilde{\mathbf{Q}}^{\text{T}} \cdot \widetilde{\mathbf{Q}}\right)^{-1}$. This proves that RDI is equivalent to DI in the case $A \leq R_Y$. In short, RDI develops through a single equation, namely (6.1): if $A > R_Y$, regularization according to (6.4) is required, while regularization is not needed if $A \leq R_Y$.

Note that if $A > R_Y$, the determination of the null space is left unchanged; therefore, (2.78) and (2.79) still apply. However, now the null space is an $(A - R_Y)$-dimensional subspace of the space of the LVs, $\mathbf{G}$ is a matrix in $\mathbb{R}^A \times \mathbb{R}^{A-R_Y}$, the columns of which are the last $A - R_Y$ left singular vectors of $\widetilde{\mathbf{Q}}^{\text{T}}$, and $\boldsymbol{\lambda}$ is a vector of arbitrary numbers in $\mathbb{R}^{A-R_Y}$. The null space uncertainty cannot be estimated in this case, as all the available approaches involve the non-invertible matrix $\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\text{T}}$. This issue will be tackled in Section 6.3.3.

RDI also generalizes the standard DI workflow applied to the case of independent quality variables, that is the case $R_Y = V_Y$ and $\text{rank}\left(\widetilde{\mathbf{Q}}^{\text{T}} \cdot \widetilde{\mathbf{Q}}\right) = \min\{A, V_Y\}$ shown in Figure 6.1(a), thus summarizing all cases discussed in Section 2.5.1. In fact, when $A \leq V_Y$ in Figure 6.1(a), $\widetilde{\mathbf{Q}}^{\text{T}} \cdot \widetilde{\mathbf{Q}}$ is a full-rank $\mathbb{R}^A \times \mathbb{R}^A$ matrix; therefore, RDI is equivalent to DI equations (2.75) and (2.76). Finally, when $A > V_Y$ in the same figure, since $\widetilde{\mathbf{Q}}^{\text{T}} \cdot \widetilde{\mathbf{Q}}$ is a matrix in $\mathbb{R}^A \times \mathbb{R}^A$ with rank V_Y, its inversion can be regularized by SVD by neglecting the last $A - V_Y$ singular values and singular vectors, which is equivalent to RDI with $R_Y = V_Y$.

In conclusions, RDI can effectively deal with the issues of collinear output variables by regularizing the matrix inversion calculations involved in DI by means of SVD. Furthermore, this methodology allows one to use a single set of equations, regardless of the relationship between A, R_Y and V_Y, thus simplifying the workflow of LVMI.

6.3.2 Regularized direct inversion for correlated quality variables

Unless some quality variables are computed as linear combinations of measured variables, the numerical rank of the $\mathbf{Y}$ matrix always matches V_Y due to measurement noise. Correlation among quality variables could exist nonetheless, therefore the true rank of $\mathbf{Y}$ could be less than the number of output variables ($R_Y < V_Y$) even if the quality variables are not collinear.

The case where $A > R_Y$ with $A < V_Y$ in Figure 6.1(b) offers a physical interpretation of the regularization by SVD performed by RDI. Recall that the rationale of RDI is to decompose $\mathbf{S} = \widetilde{\mathbf{Q}}^{\text{T}} \cdot \widetilde{\mathbf{Q}}$ by SVD and to retain exactly R_Y singular values for regularizing the inversion. Physically, this means that only the information related to the systematic relationship among the quality variables and encoded in $\mathbf{S}$ is retained, while the information related to the masking effect of noise is discarded. With reference to (6.2), the systematic information is represented by $\mathbf{\Sigma}_1$, while the effect of noise is represented by $\mathbf{\Sigma}_2$. Note that singular values in this latter matrix are not null anymore as they summarize the variability due to noise, but they are significantly smaller than singular values in $\mathbf{\Sigma}_1$. Therefore, two metrics quantifying the retained and lost information are the sum of singular values in $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$, respectively, possibly divided by the trace of $\mathbf{\Sigma}$ for normalization.

6.3.3 Regularized approach to the estimation of the null space uncertainty

Existing methods for the estimation of the null space uncertainty cannot be applied if $R_Y < V_Y$, as they all inherit the assumption of independent output variables from the original formulation of DI. Focusing on two simple analytical approaches, the method by Facco et al. (2015) involves the non-invertible matrix $\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}}$ in (2.84), while this matrix is found in (2.88) and (2.92) in the method proposed by Palací-López et al. (2019).

We therefore propose to improve these methods by employing RDI in place of DI. Starting with the method proposed by Facco et al. (2015), (6.1) is first used to compute the $\mathbf{t}_{\mathrm{des,p}}$ to be used in the estimation of the leverage, in (2.81). The confidence interval of $\mathbf{t}_{\mathrm{des,p}}$, originally computed as in (2.84), is then defined exploiting the regularization proposed by RDI:

$$\mathrm{CI}\left(\mathbf{t}_{\mathrm{des,p}}^{\mathrm{T}}\right) = \mathrm{CI}\left(\mathbf{y}_{\mathrm{des}}^{\mathrm{T}}\right) \cdot \widetilde{\mathbf{Q}} \cdot \mathbf{S}_{\mathrm{reg}}^{-1} \quad . \tag{6.5}$$

This definition holds for both the cases of collinear and correlated quality variable and draws full advantage of RDI while being based on a PLS model built on all the available quality variables, which is particularly relevant the estimation of $\mathrm{CI}\left(\mathbf{y}_{\mathrm{des}}^{\mathrm{T}}\right)$ in (2.83).

Concerning the method proposed by Palací-López et al. (2019), RDI is used to redefine (2.88) as:

$$\tilde{\mathbf{t}}_{\mathrm{des}_l}^{\mathrm{T}} = \mathbf{t}_{\mathrm{des}_l}^{\mathrm{T}} + \mathbf{r}_{\mathrm{des}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}} \cdot \mathbf{S}_{\mathrm{reg}}^{-1} \quad , \tag{6.6}$$

and (2.92) as:

$$\mathrm{CI}\left(\mathbf{t}_{\mathrm{des}_l}^{\mathrm{T}}\right) = \mathbf{t}_{\mathrm{des}_l}^{\mathrm{T}} \pm \mathbf{s}_{\hat{\mathbf{y}}_{\mathrm{des}_l}} t|_{\frac{\alpha}{2}} \cdot \widetilde{\mathbf{Q}} \cdot \mathbf{S}_{\mathrm{reg}}^{-1} \quad . \tag{6.7}$$

Both these equations are independent of the selected null space point $\mathbf{t}_{\mathrm{des}_l}$. However, note that an indirect effect exists in the case of correlated variables. If one naïvely used DI to estimate $\mathbf{t}_{\mathrm{des,p}}$ when $A > V_Y > R_Y$, numerical errors arising from the inversion of $\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}}$ in (2.77), due to the reciprocal of the small diagonal elements of $\mathbf{\Sigma}_2$ in (6.2), are expected to displace $\mathbf{t}_{\mathrm{des,p}}$ from its true position. As the method proposed by Palací-López et al. (2019) yields a "hourglass-shaped" confidence interval with minimum uncertainty at $\mathbf{t}_{\mathrm{des,p}}$, that is the same one estimated with the method by Facco et al. (2015), the hourglass could be misplaced due to the displacement of $\mathbf{t}_{\mathrm{des,p}}$. This could clearly bias the decision of process engineers when considering the uncertainty while moving the inversion solution along the null space. RDI effectively solves this issue.

6.3.4 Advantages and limitations of regularized direct inversion

In light of the interpretation of RDI outlined in the previous Sections, the advantages of the proposed approach over the existing DI approach are apparent. Compared to standard DI where one retains all quality variables in $\mathbf{Y}$, notwithstanding the fact that they may be correlated, RDI effectively solves ill-conditioning issues arising when $A > R_Y$. In particular, note that this becomes quite important when $A > V_Y$: $\widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{Q}}^{\mathrm{T}}$ in (2.77) is still an $\mathbb{R}^{V_Y} \times \mathbb{R}^{V_Y}$ matrix, thus ill-conditioned (the effect of singular values in $\mathbf{\Sigma}_2$, due to the masking effect of noise, cannot be

removed with the generalized inverse method). This point is even more important when it comes to the estimation of the null space uncertainty, as explained in Section 6.3.3.

RDI offers advantages also over the two methods originally proposed by Jaeckle et al. (1998) to address the problem of correlated output variables. One method implies removing $V_Y - R_Y$ variables from matrix **Y**. On the other hand, RDI enables including all the available quality variables in **Y**. This enables one to encode all available information in the PLS model, even when the quality variables are correlated. In fact, such correlation is exploited in PLS model calibration to define the output LVs and is encoded in matrix $\widetilde{\mathbf{Q}}$. Such additional information is also used in the computation of the PLS prediction uncertainty, which is the foundation of the analytical methods to estimate the null space uncertainty.

The alternative approach proposed by Jaeckle et al. (1998) to cope with output correlation consists in first performing a PCA on the **Y** matrix with all response variables, then using the score matrix yielded by the PCA model as quality matrix for PLS model calibration and inversion. In addition to the drawbacks already discussed in Section 6.2.2, this approach implies another subtler downside: the artificial quality variables to be used in LVMI (the scores of the PCA model on the real quality variables) are computed independently of the input variables. In a sense, this makes such an approach to LVMI similar to principal components regression (PCR; see Geladi et al. (1986) for details on PCR), and exposes it to a well-known drawback of this method. In PCR, the predictor matrix **X** is replaced by scores computed by PCA on **X**, which are computed independently of the responses. It has been proved that such an approach is suboptimal, because disregarding the aim of modeling (prediction, in the case of PCR) when encoding the predictors, can be detrimental. In fact, the scores used as predictors could encode variance relevant for the operating space but uncorrelated to the quality space (Geladi et al., 1986), therefore introducing noise components that may negatively impact on the predictive performance of the model. This in turn requires adding more PCs to "find" the variance in the operating space that is useful to predict the quality space (Wise et al., 1996). Translated to the LVMI perspective, this means that encoding the responses disregarding their relationship with predictors may be detrimental to the model inversion performance. On the other hand, building predictor LVs directly by PLS accounts for the aim of modeling and optimizes the LVs in the operating and quality spaces in light of this objective, therefore extracting the variance in **X** correlated to the variance in **Y** directly and in the first LVs. This leads to better performance in prediction (Geladi et al., 1986).

A similar reasoning could also be applied to the approach to LVMI being discussed, where output variables are first transformed by PCA independently of input variables, and only then are used to build a PLS model. One cannot know in advance if the artificial quality variables (PCA scores) are easier or harder to predict. Furthermore, if information in the quality space relevant to prediction is not captured by the first R_Y principal components of the PCA model, the prediction performance of PLS surely degrades, which in turn compromises also the model

inversion performance. Such a drawback does not affect the proposed RDI formulation. In fact, the "raw" quality variables, and all of them, are used in **Y**. Therefore, the LVs are computed directly by the PLS model, but starting from true quality variables, and they are optimized for prediction based on the operating space.

It is worth highlighting that RDI does not solve a drawback common to both the approaches proposed by Jaeckle et al. (1998): an approximation is still required in order to tackle the ill-conditioning problem. Therefore, some useful information might still be lost in regularizing the inversion by neglecting entities with subscript 2 in (6.2). However, RDI relies on a PLS model built with all response variables in **Y**, which offers the advantage of encoding all useful information about the input-output relation directly into the model. Specifically, information about output variables is collected by matrix $\widetilde{\mathbf{Q}}$. As stated in Section 6.3.2, neglecting irrelevant singular values implies neglecting only information due to effect of noise (which masks the true relationship among response variables), while retaining the systematic information. Finally, all the benefits of RDI outlined in this Section are integrally inherited by the proposed approaches to estimate the uncertainty of the null space.

6.4 Case study 1: batch fermentation

The first case study is discussed in this Section and used to compare the original DI approach, coupled to alternative approaches to accommodate the issue of output correlation (Jaeckle et al., 1998), to the proposed RDI approach in order to show the advantages of including all quality variables in **Y** even when some of them are correlated. A simulated isothermal batch fermentation process is considered. The product design task is to find the initial conditions for the batch that are required to obtain an assigned multivariate specification on the end-product, under the simplifying assumptions that no disturbances affect the process and no noise exists on the measurements. The first-principles model of the systems, described in the next Section, is used to generate the dataset for PLS model calibration and inversion; all computations are carried out using MATLAB R2021a (The Mathworks, 2021) with in-house-developed code.

6.4.1 Data generation

The process is described by means of a simplified model of a generic batch fermentation. Three species are considered in the fermenter: two substrates, denoted as S_1 and S_2, and a growth inhibitor, denoted as I_1, which is also assumed to be toxic for the biomass, causing cell death. A single product, P, is obtained. Dead cells are considered separately from viable cells. The model consists of the following differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}X_\mathrm{v} = X_\mathrm{v}\left(\mu_\mathrm{max}\frac{c_{\mathrm{S}_1}}{K_{\mathrm{s},\mathrm{S}_1}+c_{\mathrm{S}_1}}\frac{c_{\mathrm{S}_2}}{K_{\mathrm{s},\mathrm{S}_2}+c_{\mathrm{S}_2}}\frac{K_{\mathrm{i},\mathrm{I}_1}}{K_{\mathrm{i},\mathrm{I}_1}+c_{\mathrm{I}_1}} - \mu_{\mathrm{d,max}}\frac{c_{\mathrm{I}_1}}{K_{\mathrm{s},\mathrm{I}_1}+c_{\mathrm{I}_1}}\right) \quad , \tag{6.8}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}X_\mathrm{d} = X_\mathrm{v}\mu_{\mathrm{d,max}}\frac{c_{\mathrm{I}_1}}{K_{\mathrm{s},\mathrm{I}_1}+c_{\mathrm{I}_1}} \quad , \tag{6.9}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}c_{\mathrm{S1}} = -X_{\mathrm{v}}k_{\mathrm{S}_1}c_{\mathrm{S}_1} \quad , \tag{6.10}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}c_{\mathrm{S2}} = -X_{\mathrm{v}}\rho^{\mathrm{c}}_{\mathrm{S}_2,\max}\frac{c_{\mathrm{S}_2}}{K^{\mathrm{c}}_{\mathrm{s},\mathrm{S}_2}+c_{\mathrm{S}_2}} \quad , \tag{6.11}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}c_{\mathrm{I1}} = X_{\mathrm{v}}Y^{\mathrm{p}}_{\mathrm{I}_1,\mathrm{S}_1}k_{\mathrm{S}_1}c_{\mathrm{S}_1} \quad , \tag{6.12}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\omega_{\mathrm{P}} = X_{\mathrm{v}}m^{\mathrm{p}}_{\mathrm{P}} \quad , \tag{6.13}$$

where X_{v} represents the concentration of viable cells and X_{d} denotes the concentration of dead cells, both in Mcell L^{-1} (million cells per liter), while c_i is the concentration of species i in mmol L^{-1}, and ω_{P} is the product concentration in mg L^{-1}. The model parameters are listed in Table 6.2, together with their definitions and nominal values.

Table 6.2. *Case study 1. Parameters of the model reported in equations (6.8) to (6.13).*

Parameter	Definition	Value	Units
$\mu_{\max}$	Maximum biomass specific growth rate	0.03836	h^{-1}
$K_{\mathrm{s},\mathrm{S}_1}$	Half-saturation constant of S_1 for biomass growth	3.172	mmol L^{-1}
$K_{\mathrm{s},\mathrm{S}_2}$	Half-saturation constant of S_2 for biomass growth	0.01	mmol L^{-1}
$K_{\mathrm{i},\mathrm{I}_1}$	Inhibition constant of I_1 for biomass growth	100	mmol L^{-1}
$\mu_{\mathrm{d,max}}$	Maximum biomass specific death rate	0.02511	h^{-1}
$K_{\mathrm{s},\mathrm{I}_1}$	Half-saturation constant of I_1 for biomass death	62.31	mmol L^{-1}
k_{S_1}	First-order constant for consumption of S_1	$1.682 \cdot 10^{-5}$	mmol Mcell^{-1} h^{-1}
$\rho^{\mathrm{c}}_{\mathrm{S}_2,\max}$	Maximum S_2 specific consumption-rate	$6.177 \cdot 10^{-4}$	mmol Mcell^{-1} h^{-1}
$K^{\mathrm{c}}_{\mathrm{s},\mathrm{S}_2}$	Half-saturation constant of S_2 for S_2 consumption	1.15	mmol L^{-1}
$Y^{\mathrm{p}}_{\mathrm{I}_1,\mathrm{S}_1}$	Yield coefficient of S_1 for I_1 production	1.531	$(-)$
$m^{\mathrm{p}}_{\mathrm{P}}$	Zero-order product specific secretion-rate	$3.698 \cdot 10^{-4}$	mg Mcell^{-1} h^{-1}

The model reported in equations (6.8) to (6.13) is a continuous-time state-space model comprising 6 state variables (X_{v}, X_{d}, c_{S1}, c_{S2}, c_{I1}, and ω_{p}). The initial values of 4 of them (X_{v}, c_{S1}, c_{S2}, and c_{I1}) are used to build the predictor matrix **X**; the response matrix **Y** may comprise only one or both end-point values for the two of the states (X_{v} and ω_{p}). 51 different initial conditions for the 6 state variables are obtained by sampling them within the intervals listed in Table 6.3. Quasi-random sampling is performed using a Sobol sequence by virtue of its space-filling properties (Garud et al., 2017). Each batch is simulated for a fixed duration of 200 h.

6.4.2 Model calibration and inversion

A graphical representation of the available dataset is provided in Figure 6.2: initial conditions and end-point quality of the historical batches are reported in Figure 6.2(a) and Figure 6.2(b), respectively. Figure 6.2(b) clarifies that the end-point quality variables are correlated, yet not

Table 6.3. *Case study 1. Lower and upper bounds of the intervals used for sampling the initial conditions of the batches in order to build the historical dataset.*

Variable	Lower bound	Upper bound	Unit
X_{v}	70	150	Mcell L^{-1}
X_{d}	0	10	Mcell L^{-1}
c_{S1}	32	60	mmol L^{-1}
c_{S2}	5	22	mmol L^{-1}
c_{I1}	0	6	mmol L^{-1}
ω_{P}	0	0.002	mg L^{-1}

collinear (the sample correlation coefficient is 0.90), and the conditions of this case study are therefore those of Figure 6.1(b). Therefore, although the numerical rank of $\mathbf{Y}$ is 2 ($V_Y = 2$), the true rank of the matrix is $R_Y = 1$. One of the observations in the historical dataset, namely the one corresponding to the end-point quality $\mathbf{y}_{\mathrm{des}} = [362.4\ \mathrm{Mcell\ L^{-1}} \quad 41.5\ \mathrm{mg\ L^{-1}}]^{\mathrm{T}}$, is selected as the product quality to be achieved in the process; this observation is therefore removed from the predictor and response matrices, leaving 50 observations to use as the calibration dataset.

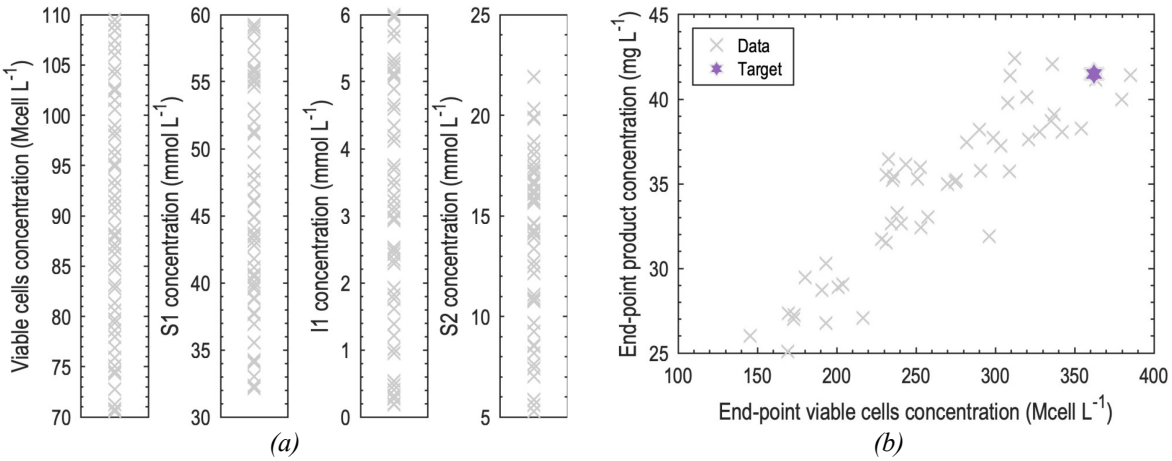


Figure 6.2. *Case study 1. Available data for (a) initial conditions and (b) end-point quality of the batches in the historical dataset.*

Three different PLS models are built using the calibration dataset. While the predictor matrix is the same in all models, that is $\mathbf{X} \in \mathbb{R}^{50} \times \mathbb{R}^{4}$, the response matrix is different for each model, depending on how the output correlation issue is tackled in model inversion.

Model 1 Only one of the output variables is included in the response matrix, selected according to the procedure suggested by Jaeckle et al. (1998), and the resulting response matrix is denoted as $\mathbf{Y}^{\mathrm{I}} \in \mathbb{R}^{50} \times \mathbb{R}^{1}$. The selected variable is the product concentration.

Model 2 The matrix including both output variables is decomposed by PCA, and the scores on the first principal component only (explaining 95% of the variability of the data) are used to build the response matrix (Jaeckle et al., 1998). This matrix is denoted as $\mathbf{Y}^{\mathrm{II}} \in \mathbb{R}^{50} \times \mathbb{R}^{1}$.

Model 3 Both quality variables are retained in the response matrix (according to the proposed RDI approach). This matrix is denoted as $\mathbf{Y}^{\mathrm{III}} \in \mathbb{R}^{50} \times \mathbb{R}^{2}$.

Note that, in all models, the true rank of the response matrix is $R_Y = 1$. To allow for a fair comparison, the number of LVs is selected *a priori* to be the same in all models; namely, $A = 2$ is selected. Therefore, a one-dimensional null space exists in all models. Table 6.4 summarizes the explained variances (EVs) of the three PLS models.

Table 6.4. *Case study 1. Diagnostics of the three PLS models built to compare different model inversion approaches. EV_X and EV_Y are the explained variances of predictors and responses, respectively.*

	Model 1		**Model 2**		**Model 3**	
LV	EV_X	EV_Y	EV_X	EV_Y	EV_X	EV_Y
1	0.2717	0.9317	0.2970	0.9282	0.2979	0.8801
2	0.2802	0.0579	0.2555	0.0631	0.2515	0.0953
Total	0.5519	0.9895	0.5525	0.9913	0.5494	0.9754

The unscaled nominal target quality $\mathbf{y}_{\mathrm{des}}$ is transformed according to the rationale used in each of the three PLS models described above, thus obtaining three different target vectors:

- the target quality to be used in the inversion of Model 1, $\mathbf{y}_{\mathrm{des}}^{\mathrm{I}} \in \mathbb{R}$, is obtained by using only the product concentration, therefore $\mathbf{y}_{\mathrm{des}}^{\mathrm{I}} = 41.5 \text{ mg L}^{-1}$;
- the target quality to be used in the inversion of Model 2, $\mathbf{y}_{\mathrm{des}}^{\mathrm{II}} \in \mathbb{R}$, is obtained by applying to $\mathbf{y}_{\mathrm{des}}$ the PCA model developed on $\mathbf{Y}^{\mathrm{II}}$, which yields $\mathbf{y}_{\mathrm{des}}^{\mathrm{II}} = 2.17$ (recall this is the value of a PCA score);
- the target quality to be used in the inversion of Model 3, $\mathbf{y}_{\mathrm{des}}^{\mathrm{III}} \in \mathbb{R}^{2}$, is the same as $\mathbf{y}_{\mathrm{des}}$, therefore $\mathbf{y}_{\mathrm{des}}^{\mathrm{III}} = [362.4 \text{ Mcell L}^{-1} \quad 41.5 \text{ mg L}^{-1}]^{\mathrm{T}}$.

Four different approaches to the solution of the model inversion problems are considered:

a. DI of Model 1 by means of (2.77);
b. DI of Model 2 by means of (2.77);
c. DI of Model 3 by means of (2.76) (note that ill-conditioning issues are expected in this case, due to correlation between output variables);
d. RDI of Model 3 by means of (6.1) retaining only $R_Y = 1$ singular values to regularize the inversion by SVD as in (6.4).

Figure 6.3(a) shows a detailed view of the designed initial conditions obtained by PLS model inversion according to each of the four approaches described above, together with the

calibration data closer to them. Figure 6.3(b) reports a detailed view of the target end-point quality (and the calibration data closer to it), together with the end-point quality actually achieved by running the process with the corresponding designed initial conditions.

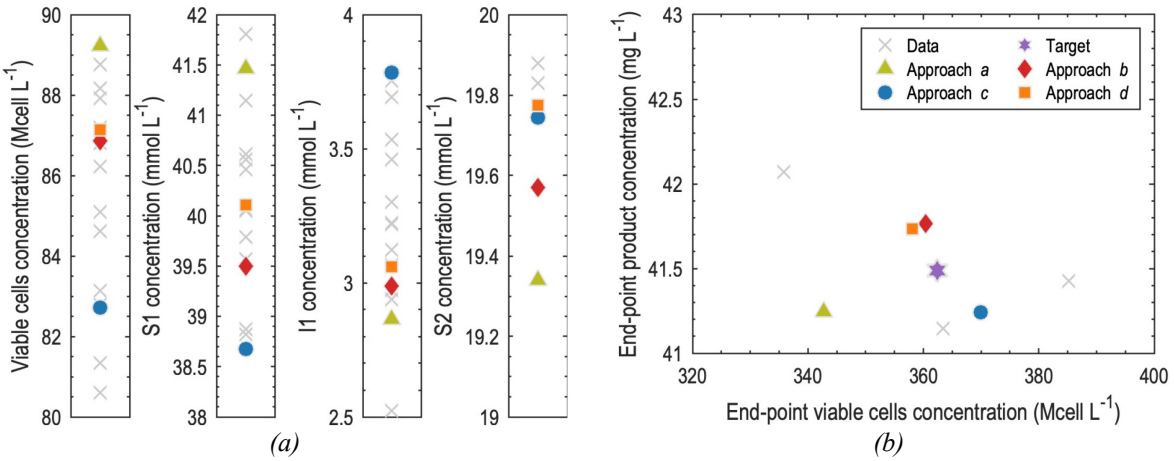


Figure 6.3. *Case study 1. Results of PLS model inversion for A = 2 latent variables using the four approaches described in Section 6.4.2. (a) designed initial conditions, and (b) end-point quality achieved by running the process at the designed initial conditions, compared to the target quality.*

It appears from Figure 6.3(b) that the end-point product concentration achieved with all four model-inversion strategies is roughly the same. However, the quality yielded by approach *a* is off-target in terms of end-point viable cells concentration. This is because approach *a* applies DI to Model 1, which does not consider the end-point viable cells concentration as an output variable to be included in $\mathbf{Y}$. Therefore, omitting quality variables from $\mathbf{Y}$ may not be the best option in product design, even if the quality variables that are omitted are strongly correlated to those that are included in $\mathbf{Y}$.

The information loss implied by neglecting one of the two quality variables can be quantified as suggested by Yacoub et al. (2004). Given the strong correlation between quality variables, it appears reasonable to establish a linear regression model $\widehat{\mathbf{Y}}^{\mathrm{III}} = \mathbf{Y}^{\mathrm{I}} \cdot \boldsymbol{\beta}^{\mathrm{T}}$ to predict $\mathbf{Y}^{\mathrm{III}}$ from $\mathbf{Y}^{\mathrm{I}}$, the columns of which are a subset of the columns of $\mathbf{Y}^{\mathrm{III}}$. After autoscaling of the two matrices, the regression coefficients are estimated as:

$$\boldsymbol{\beta}^{\mathrm{T}} = ((\mathbf{Y}^{\mathrm{I}})^{\mathrm{T}} \cdot \mathbf{Y}^{\mathrm{I}})^{-1} \cdot (\mathbf{Y}^{\mathrm{I}})^{\mathrm{T}} \cdot \mathbf{Y}^{\mathrm{III}} \quad . \tag{6.14}$$

The coefficient of determination, R^2, is then used as a measure of the information on the end-point viable cells concentration that can be represented (meaning, predicted) using data of the final product concentration. In the case of end-point viable cells concentration predicted by linear regression from end-point product concentration, one gets $R^2 = 0.81$, with an overall R^2 for both variables equal to 0.905 (obviously, the R^2 of final product concentration predicted from end-point product concentration is 1). This means that retaining only the selected quality variable for PLS model inversion implies a loss of 9.5% of the information on the quality space, which can impact the achieved end-point quality as seen in Figure 6.3(b).

The information loss implied by approach *a* justifies why the quality actually obtained is the farthest from the target. All other approaches are basically equivalent in terms of achieved end-point quality. Considering approach *b*, the information retained can be quantified as the variance explained by the first (and only) PC of the PCA model of $\mathbf{Y}$, which is 95.0%. This implies that only 5.0% of the information on the quality space is lost using approach *b*. On the other hand, the information loss in approach *d* can be quantified as the trace of $\mathbf{\Sigma}_2$ (singular values of $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$ of Model 3 neglected in regularization) divided by the trace of $\mathbf{\Sigma}$. In the case being considered, $\mathbf{\Sigma} = \mathrm{diag}([1.599 \quad 0.0716])$, and only the first singular value is retained to compute $\mathbf{S}_{\mathrm{reg}}^{-1}$; therefore, only 4.3% of the information about the quality space is lost in regularization. Since this value is very similar to the information lost in approach *b*, it is reasonable that approaches *b* and *d* yield comparable results. However, the advantage of approach *d* is that it deals with true quality variables, not with artificial ones (PCs).

Despite being very similar to the quality obtained with approaches *b* and *d*, the outcome of approach *c* deserves discussion under a different perspective. In principle, no information about the quality space is lost using this approach, because it retains both quality variables in $\mathbf{Y}$. However, approach *c* does not regularize the inversion. Figure 6.3(a) shows quite a difference in the initial conditions designed by approach *c* with respect to the initial conditions designed by approaches *b* and *d*, especially for the concentration of viable cells, of S_1 and of I_1. The model scores of Model 3, as plotted in Figure 6.4, help understanding why.

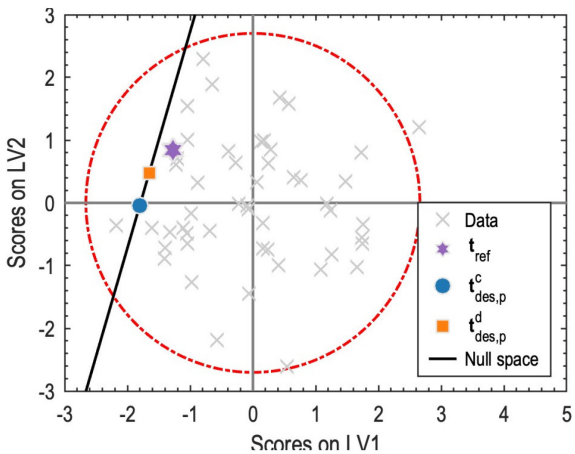


Figure 6.4. *Case study 1. Score plot of Model 3 (A = 2 latent variables). Scores obtained by direct inversion (approach c in Section 6.4.2) and by regularized direct inversion (approach d in Section 6.4.2) are shown against the target scores and the calibration scores, together with the one-dimensional null space.*

Approaches *c* and *d* both lead to a $\mathbf{t}_{\mathrm{des,p}}$ falling on the null space, as expected. However, the RDI used in approach *d* explicitly considers the presence of such null space by regularizing the inversion of matrix $\widetilde{\mathbf{Q}}^{\mathrm{T}} \cdot \widetilde{\mathbf{Q}}$, while the DI used in approach *c* does not. This exposes DI to ill-conditioning issues, therefore to the propagation of numerical errors in matrix inversion. As a consequence, in approach *c*, $\mathbf{t}_{\mathrm{des,p}}$ could in principle fall anywhere along the null space. Though

this might not be seen as an issue, given the fact that (according to the model) any set of initial conditions projecting onto the null space yields the same quality, one should bear in mind that the location of the null space gets more and more uncertain as the null space moves away from the region wherein the calibration data are available (Palací-López et al., 2019; Tomba et al., 2014); see Figure 2.7(b) for an example. Therefore, the probability for the DI solution to not fall exactly onto the true null space (hence, to obtain off-target quality) increases while moving away from the average conditions of the available data (for example, the null space uncertainty intervals at the coordinates of the solutions *c* and *d* in Figure 6.4, estimated as in Tomba et al. (2012a), have widths 5.9 and 6.3, respectively). Furthermore, designed process conditions can deviate significantly from the historical ones.

6.5 Case study 2: fed-batch penicillin manufacturing

The second case study is based on a fed-batch penicillin manufacturing process simulated through the PenSim simulator (Birol et al., 2002). The product design task is more complex than the one of the previous case study: finding the time profiles of a set of process variables that can lead to an assigned end-point quality target, under significant process variability and measurement noise. The purpose of this case study is to investigate the impact of the PLS model inversion approach (DI or RDI) on the uncertainty of the inversion results when the end-point quality variables are correlated. The uncertainty in the estimation of the null space is considered as well. MATLAB R2021a (The Mathworks, 2021) was used to carry out the computations by means of in-house-developed code.

6.5.1 Data generation

The PenSim simulator (Birol et al., 2002) implements a nonlinear state-space model with 9 states and 7 inputs (Table 6.5); 4 inputs are manipulated by the control system, while the remaining 3 are set to nominal values and kept constant during a batch. The control system is centered on two proportional-integral-derivative (PID) controllers that control the fermenter pH and temperature by manipulating the acid/base feed flow rates and cooling/heating water flow rates, respectively. The model simulates realistic process variability and measurement noise. The reader is referred to Birol et al. (2002) for a detailed description of the simulator.

The simulator is used to simulate 170 batches by changing the initial conditions and the nominal inputs as indicated in Table 6.5. Namely, the initial conditions for penicillin concentration and generated heat are always set to 0. The initial conditions for the remaining state variables are sampled from independent normal distributions. The nominal values of the 3 simulator inputs not manipulated by a controller are sampled by independent normal distributions. The remaining inputs are manipulated by the controllers with default tuning. All model parameters are set to default values, as assigned by Birol et al. (2002).

Table 6.5. *Case study 2. State and input variables of the PenSim simulator (Birol et al., 2002) with initial conditions for the states and nominal values of the inputs to the simulator, as used to generate the historical dataset. The notation $\mathcal{N}(\mu, \sigma^2)$ means the value is sampled from a normal distribution with mean μ and variance σ^2. PID means that the input is manipulated by a proportional-integral-derivative controller.*

Variable	Initial conditions or nominal values	Unit
Simulator states		
Substrate concentration	$\mathcal{N}(15, 1.5^2)$	g L^{-1}
Dissolved oxygen	$\mathcal{N}(1.16, 0.008^2)$	mmol L^{-1}
Biomass concentration	$\mathcal{N}(0.1, 0.2^2)$	g L^{-1}
Penicillin concentration	0	g L^{-1}
Culture volume	$\mathcal{N}(100, 4^2)$	L
Dissolved carbon dioxide	$\mathcal{N}(0.5, 0.008^2)$	mmol L^{-1}
H^+ concentration	$\mathcal{N}(10^{-5}, (0.1^{-5})^2)$	mmol L^{-1}
Fermenter temperature	$\mathcal{N}(298, 0.3^2)$	mol L^{-1}
Generated heat of reaction	0	cal
Simulator inputs		
Air feed flow rate	$\mathcal{N}(8.6, 0.3^2)$	L h^{-1}
Stirring power	$\mathcal{N}(30, 3^2)$	W
Substrate feed flow rate	$\mathcal{N}(0.042, 0.0015^2)$	L h^{-1}
Acid feed flow rate	PID	L h^{-1}
Base feed flow rate	PID	L h^{-1}
Cooling water flow rate	PID	L h^{-1}
Heating water flow rate	PID	L h^{-1}

Each batch is simulated for a fixed duration of 300 h. We assume that 12 variables can be measured in real time (Table 6.6), with a sampling interval of 1 h and default noise level. Their time profiles are arranged in a tensor to characterize the process operation, with the batches in different rows, measurements in different columns, and time along the third dimension. BWU of this matrix (Nomikos et al., 1995b) is used to build the regressor matrix **X**, the columns of which therefore include a total of $V_X = 12 \cdot 300 = 3600$ pseudo-variables. End-point values of the biomass and penicillin concentrations are recorded to characterize the end-of-batch quality. Their values are arranged in the response matrix **Y** ($V_Y = 2$).

6.5.2 Model calibration and inversion

The product design task is finding the time profiles of the PLS model predictors that enable achieving the target end-point product quality $\mathbf{y}_{\text{des}} = [13.4 \text{ g L}^{-1} \quad 1.40 \text{ g L}^{-1}]^{\text{T}}$, where the

Table 6.6. *Case study 2. Variables included in the PLS model predictor matrix* ***X*** *(time profiles along the batch) and in the response matrix* ***Y*** *(end-point values only).*

Designation	Variable
Predictors (time profiles)	Substrate concentration
	Dissolved oxygen
	Culture volume
	Dissolved carbon dioxide
	Fermenter pH
	Fermenter temperature
	Air feed flow rate
	Stirring power
	Substrate feed flow rate
	Acid feed flow rate
	Base feed flow rate
	Cooling water flow rate
Responses (end-of-batch values)	Biomass concentration
	Penicillin concentration

first element refers to the biomass concentration and the second one to the penicillin concentration. The target product is obtained from one of the simulated batches, and the corresponding observation is therefore removed from the calibration dataset. Note that, in a real setting, not all the time profiles obtained through model inversion can be assigned for operating the fermenter, because the number of variables that can be manipulated directly (or assigned as controller set-points) are only a subset of those included in **X**. Therefore, one may assign the trajectories of this subset of variables only, and then assess in real time whether the batch is evolving according to the expectations or not by comparing the actual time evolution of the remaining subset of variables against their designed trajectories. Alternatively, advanced model based-control approaches in the latent-variable space can be used (Golshan et al., 2011). Yet, in this case study we do not address this issue explicitly, but we explore a different aspect of LVMI, namely the uncertainty of the time profiles designed by model inversion.

When using LVMI to design the time profiles of the regressors (namely, of the fermenter operating conditions), one issue to consider is understanding how much variability can be afforded in the designed process conditions in order to keep the end-point quality reasonably close to the target, regardless of the process control system that might be in place. As the aim is understanding how the designed process conditions change in response to a variation of the assigned quality around the target one, one way for assessing this is to sample batches around $\mathbf{y}_{\mathrm{des}}$ and perform PLS model inversion for each batch. This study adopts this approach because

of its simplicity and minimal computational burden, but other approaches to evaluate the uncertainty of the solution of the inversion problem exist (Bano et al., 2018b; Wang et al., 2020). 100 sample batches are drawn around $\mathbf{y}_{\text{des}}$; the two quality variables are sampled from independent normal distributions, namely $\mathcal{N}(13.4, 0.1^2)$ for the biomass concentration, and $\mathcal{N}(1.40, 0.01^2)$ for the penicillin concentration. The variances for sampling are selected in order to yield a reasonable variability around the target value, while not breaking the correlation structure of observations in the historical data (Jaeckle et al., 2000). The quality data for calibration, target end-product, and sample batches around it are shown in Figure 6.5.

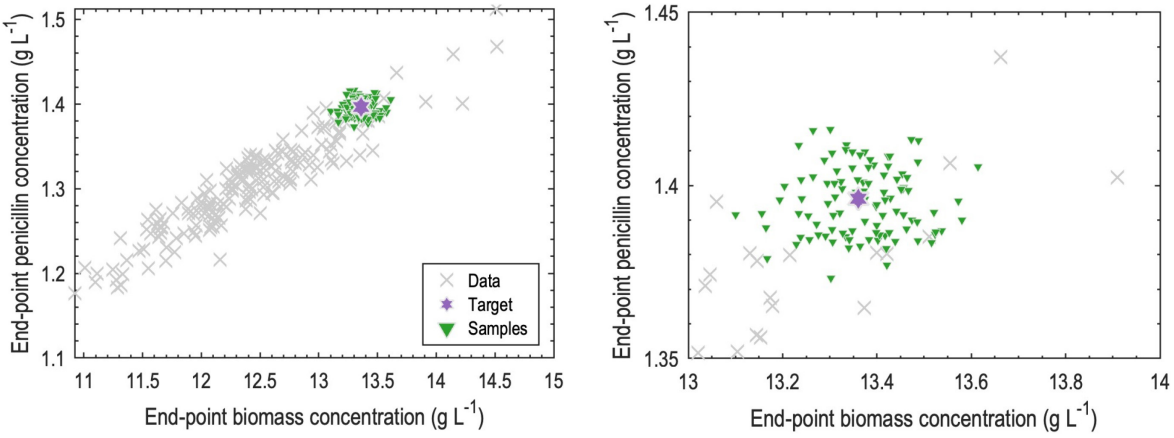


Figure 6.5. *Case study 2. Calibration quality data, quality target, and sample batches around the target: (a) full-scale and (b) detail of sample batches for variability estimation.*

Figure 6.5(a) indicates that the biomass concentration and penicillin concentration are strongly correlated (sample correlation coefficient: 0.93). Therefore, also for this case study the conditions illustrated in Figure 6.1(b) hold true. The numerical rank of $\mathbf{Y}$ is 2 ($V_Y = 2$), but its actual rank is $R_Y = 1$. We analyze the effect of two different choices of the number of LVs, A, in such a way as to study the impact on the uncertainty of the model inversion results of different relationships between A, V_Y, and R_Y. Table 6.7 summarizes the diagnostics of the PLS models for $A = 1$ and $A = 2$.

Table 6.7. *Case study 2. Diagnostics of the PLS model. EV_X and EV_Y are the explained variances of predictors and responses, respectively.*

LV	$\mathbf{EV}_X$	$\mathbf{EV}_Y$
1	0.2330	0.8952
2	0.1007	0.0423
Total	0.3337	0.9375

First, a PLS model with $A = 1$ is considered; hence, $A = R_Y$ in this case. In Figure 6.6(a), the model scores are shown against one of the quality variables (due to strong correlation, the plot

for the other one is qualitatively the same). Namely, the scores of the historical products are shown together with the true (and unknown) score of the target product ($\mathbf{t}_{\text{ref}}$), and with $\mathbf{t}_{\text{des}}$ as obtained by RDI of the model. Figure 6.6(b) shows the time profile for one process operating condition (acid feed flow rate) as obtained by RDI, along with an estimation of its variability across the sample batches; the profiles of the same variable in the calibration dataset are also plotted for comparison. Results for DI are not shown because they are identical to those for RDI, as expected since $A = R_Y$, and (2.75) can therefore be used for model inversion.

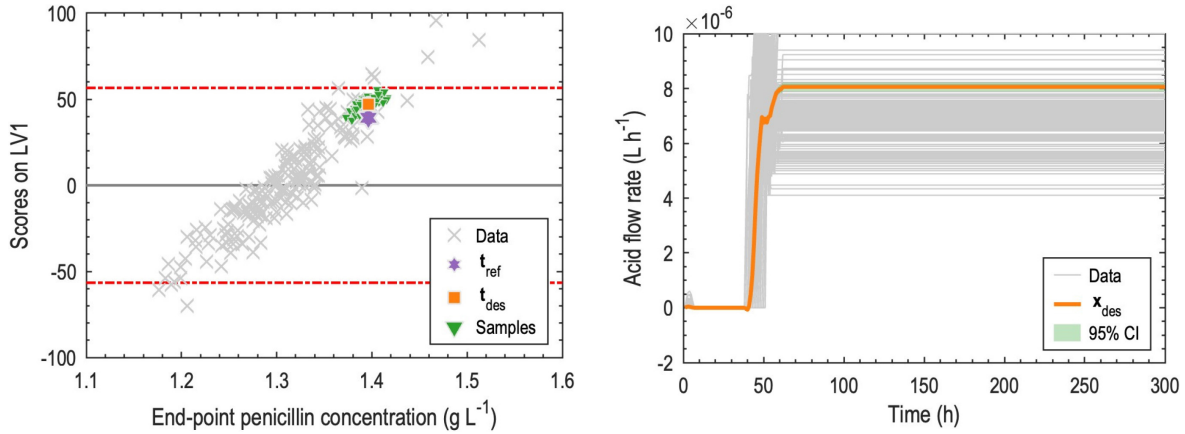


Figure 6.6. *Case study 2. Results of PLS model inversion for A = 1 latent-variable. (a) Scores obtained by RDI for one of the quality variables are shown against the target score, the scores of historical observations, and the scores of the sample batches drawn around the target; the dash-dotted lines bound the 95% confidence region for the historical scores. (b) Designed time profile for one process operating condition with estimated 95% confidence interval (CI), compared to the time profiles of the observations in the calibration dataset.*

Figure 6.6(a) clearly shows that all sample batches drawn around $\mathbf{y}_{\text{des}}$ are projected onto the single LV of the model properly, close to the designed $\mathbf{t}_{\text{des}}$. Similarly, Figure 6.6(b) tells us that the variability of the designed time profile of the acid flow rate (which is barely visible) is much smaller than the variability of the calibration data, as one would indeed desire in a product design exercise. Therefore, in this case, DI and RDI lead to equivalent results in terms of both designed process conditions and estimation of their variability.

Next, we consider a PLS model with $A = 2$ LVs to highlight the advantages of the proposed RDI approach with respect to DI. In this case, $A = V_Y$, hence $A > R_Y$. Let us consider DI first. Given that the quality variables are correlated, one may consider removing one variable from $\mathbf{Y}$, but (as seen for the previous case study) this may preclude achievement of the target for the quality variable that is not included in the response matrix. On the other hand, if both quality variables are included in $\mathbf{Y}$, (2.76) requires inverting a matrix in $\mathbb{R}^2 \times \mathbb{R}^2$ with rank $R_Y = 1$, which gives rise to numerical issues. In fact, Figure 6.7(a) shows that sample batches projected onto the score space get scattered along the null space, which has dimension $A - R_Y = 1$. This causes the variability of the designed process conditions to be largely overestimated, as shown

in Figure 6.7(b): the estimated variability of the designed acid flow rate profile is comparable to the variability of the entire set of calibration data, thus making the inversion results unreliable. Note that, since all sampled batches used for variability estimation feature a different quality, it makes no sense that samples scatter along the null space, as that direction should cause no quality variation.

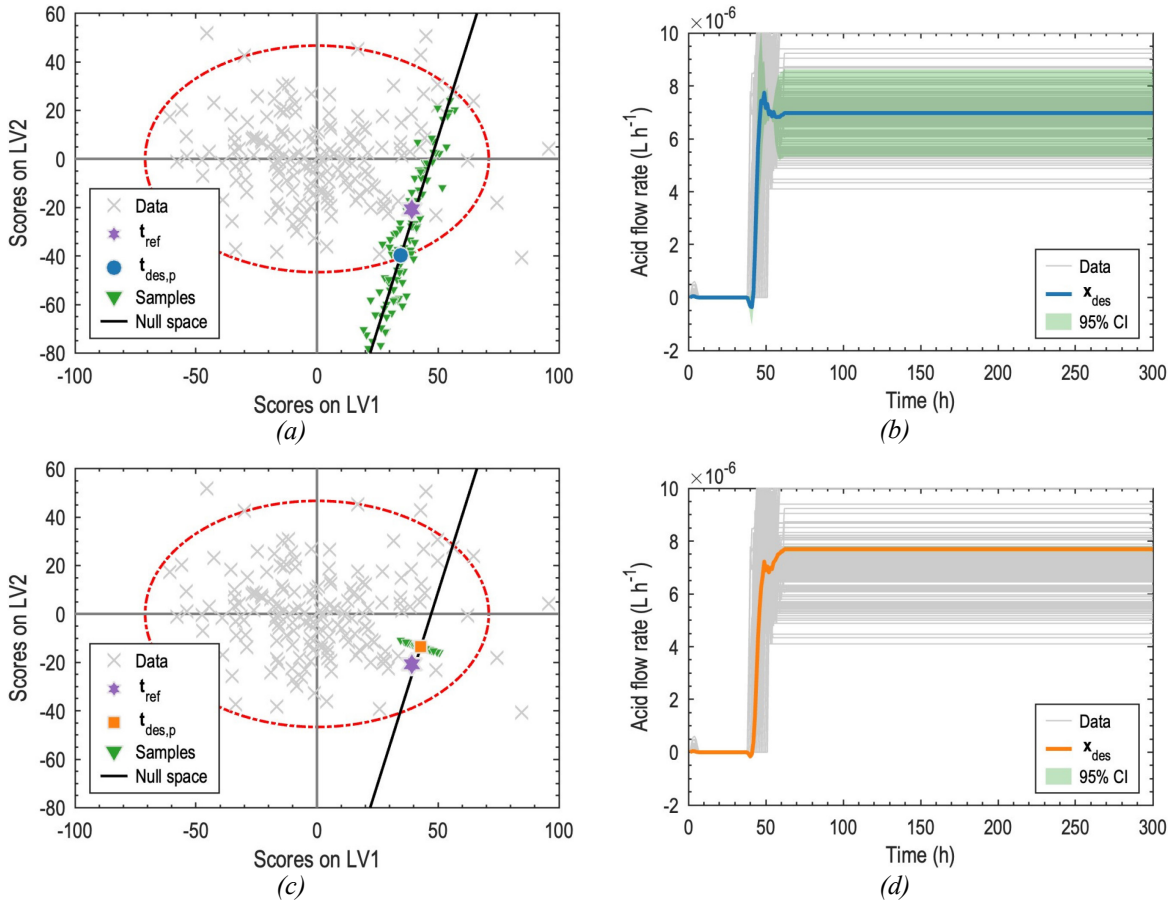


Figure 6.7. *Case study 2. Results of PLS model inversion for A = 2 latent variables; a one-dimensional null space exists (thick black line in the left plots). (a) Direct inversion: scores of the process conditions for the target, designed, calibration, and sample batches. (b) Direct inversion: designed time profile for one process condition with estimated 95% confidence interval (CI), compared to observations in the calibration dataset. (c) Regularized direct inversion: scores of the process conditions for the target, designed, calibration, and sample batches. (d) Regularized direct inversion: designed time profile for one process condition with estimated 95% confidence interval (CI), compared to observations in the calibration dataset.*

Results are significantly different for the proposed RDI approach, which allows to include both quality variables in $\mathbf{Y}$ without any numerical issues, provided that only one singular value is retained in the matrix inversion operations involved in (6.1) and (6.4). The sample batches around $\mathbf{y}_{\text{des}}$ are in fact projected onto the scores space very close to $\mathbf{t}_{\text{des,p}}$ in Figure 6.7(c), and the designed process conditions are subject to a very small variability in Figure 6.7(d). Therefore, the model inversion results can be deemed reliable. Also note from Figure 6.7(c) that the sample batches correctly scatter across, and not along, the null space, thus indicating

that they are representing the variation of the quality in a direction that is orthogonal to the null space, that is the direction in the latent space truly describing quality.
Results for a PLS model built on $A = 3$ LVs are qualitatively similar to those for $A = 2$, and are therefore omitted for brevity.

6.5.3 Estimation of null space uncertainty

The estimation of the uncertainty of the null space is discussed to conclude the second case study. We consider the PLS model with $A = 2$ introduced in the previous Section, and we use the analytical approach proposed by Palací-López et al. (2019) for null space uncertainty estimation, although the discussion holds for other methods as well.
If one uses DI in this case, Figure 6.7(a) clearly shows the propagation of the numerical errors due to the inversion of an ill-conditioned matrix, which causes the sample batches to scatter along the null space and inflates the uncertainty estimated on the designed process conditions, as in Figure 6.7(b). As the same ill-conditioned matrix is to be inverted to estimate the uncertainty of the null space, a similar inflation is to be expected. This hypothesis is confirmed by Figure 6.8, which compares the scores from DI and RDI, $\mathbf{t}_{\text{des}}^{\text{DI}}$ and $\mathbf{t}_{\text{des}}^{\text{RDI}}$, respectively, with the estimated null space confidence interval estimated with the DI-based approach proposed by Palací-López et al. (2019), $\text{CI}\left(\mathbf{t}_{\text{des}}^{\text{DI}}\right)$, and the one computed by the RDI-based method proposed in this Chapter, $\text{CI}\left(\mathbf{t}_{\text{des}}^{\text{RDI}}\right)$.

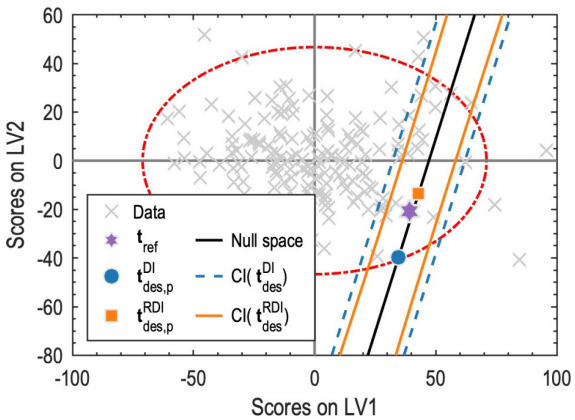


Figure 6.8. *Case study 2. Comparison of the approaches for estimation of the null space uncertainty in the score space of the PLS model with A = 2. Calibration and target scores are shown together with the ones designed with DI and RDI. A one-dimensional null space exists (thick black line), shown with the confidence intervals (CIs) based on DI (dashed lines) and RDI (solid orange lines).*

Figure 6.8 shows that the proposed method correctly estimates the null space uncertainty, while the DI-based approach yields an inflated estimate. Furthermore, the improved accuracy of the RDI solution ($\mathbf{t}_{\text{des}}^{\text{RDI}}$ is closer to the reference value than $\mathbf{t}_{\text{des}}^{\text{DI}}$) allows to correctly position the point of minimal uncertainty of the "hourglass", which should in principle fall onto $\mathbf{t}_{\text{ref}}$. This feature of the proposed approach was conjectured in Section 6.3.3 and is proved by Figure 6.8.

6.6 Conclusions

LVMI can be exploited to perform product design, namely to find the raw materials and operating conditions (model inputs) that are required to obtain a new product with assigned quality specifications (model outputs). The only requirements are the availability of appropriate process and quality data to develop a latent-variable model, and that the target quality is consistent with the correlation structure of the historical quality data. In this Chapter, we addressed the product design problem using PLS model inversion. We proposed a novel algebraic formulation of the model inversion problem, RDI, which enables one to perform model inversion in a straightforward way also in the presence of correlated (or even collinear) outputs. The formulation is based on the SVD of the matrix to be inverted, where the decomposition factors are truncated in such a way as to retain only the systematic variability of the historical data. Furthermore, we extended available methods to estimate the uncertainty of the null space, a subspace of the space of LVs that may arise in LVMI, using RDI. The proposed formulations were successfully tested on two simulated case studies related to digital product design biochemical processes.

The most popular approach currently used to cope with output correlation in the model inversion task relies on removing some quality variables from the output matrix, and on relating only the remaining output subspace to the input space through PLS regression. However, in this case the solution obtained by model inversion may not be able to ensure that the quality variables which have not been included in the output matrix will be close enough to their targets. Conversely, the proposed RDI formulation enables one to retain in the model response matrix all quality variables (hence, to model the entire quality space), and addresses output correlation by removing *a posteriori* only the non-systematic information that would cause singularity of the matrix to be inverted. This approach yields a more reliable solution to the inversion problem as compared to removing *a priori* some quality variables, because no structural information about the relationship between inputs and outputs is left out of the model by design. Furthermore, the use of SVD provides a metric to calculate the amount of information lost in matrix regularization, which is helpful to assess the effectiveness of the inversion task. Additionally, the proposed formulation simplifies the model inversion workflow, because it develops independently of the relation between the number of selected latent variables and the rank of the output matrix. Finally, the RDI approach provides similar results to one where the dataset including all historical quality variables is first modeled through PCA, and then a subset of the PCA model scores is used to build the output matrix of the PLS model. However, RDI uses the true quality variables, and not the scores, to build the output matrix, which is much more convenient from a practical point of view.

Chapter 7

Smart process analytics for process monitoring[10]

A novel framework for data-driven fault detection in manufacturing processes is proposed in this Chapter. The framework automatically selects and calibrates the best fault detection model for a given dataset based on preliminary assessment of data characteristics and a rigorous model selection routine. We demonstrate the effectiveness of the framework on four case studies.

7.1 Introduction

Maintaining high product quality is a key requirement in manufacturing processes. The operation of the process and its effect on product quality can be supervised by process monitoring schemes (Chiang et al., 2001; Kourti, 2003; Reis et al., 2017). Fault detection is the first step in a chain of operations in process monitoring that are performed in order to recover a process to normal operating conditions, in case any fault occurs (Figure 7.1). After a fault is detected, the process/product quality variables most related to the malfunction are identified. The nature and, possibly, the root cause of the fault are then diagnosed leveraging expert process knowledge; alternatively, a classification approach can be used to diagnose the fault searching through a library of known faults. Finally, measures to recover the process operation are taken (Chiang et al., 2001). Implementing this workflow in industrial environments is of paramount importance to guarantee a consistent and on-specification product quality, thus the economic effectiveness of the production. Process monitoring is fundamental for safety as well, as testified by some recent catastrophic events where the process drifted out of control due to inappropriate monitoring (Pallardy, 2023; Saleh et al., 2014).

There is no uniform terminology in the process monitoring literature. In this Chapter, we adopt the terminology and definitions described by Raich et al. (1996). A fault is defined as «an unpermitted deviation of at least one characteristic property of a variable from an acceptable behavior» (Isermann, 2005). Therefore, a fault can be defined as an abnormal process operation regardless of whether it is caused by faulty equipment or a significant disturbance acting on the process. An example of faulty equipment is fouling in a heat exchanger that causes a notable

[10] Part of the research discussed in this Chapter is included in a manuscript in preparation (Mohr et al., 2023), to be submitted for publication as a journal paper. An open-source software package implementing the methods described herein will be made available upon publications of the paper.

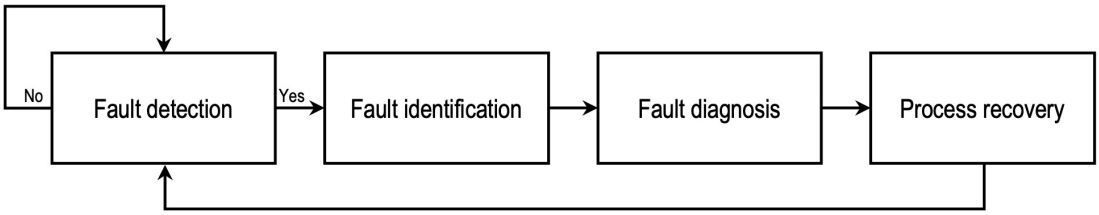


Figure 7.1. *The four steps of the process monitoring and recovery chain (Chiang et al., 2001).*

temperature difference of the outlet stream due to poor heat transfer. Alternatively, biased sensors or a sticking valve can also be considered faulty equipment. An example of a significant disturbance is a raw material that is supplied from a different provider and contains more impurities than expected, leading to an altered composition of the product (Chiang et al., 2001).
Process monitoring methods generally belong to one of the following categories: data-driven, analytical, and knowledge-based (Isermann, 1994). Data-driven methods, which are the preferred ones in the (bio-)chemical industry, are considered in this Chapter. For fault detection, the first step is to calibrate a model that describes data from the NOC of the process. Afterwards, statistical measures are used to understand whether new collected data deviate significantly from the data used in the NOC model (Qin, 2003). A significant deviation is interpreted as an indicator of a fault occurring in the process.
The data-driven approach requires the selection and calibration of a modeling method. However, several methods are available, and no method performs best on all possible problems. Very few individuals possess significant expertise on a large number of fault detection methods that can provide strong performance, and practitioners usually select the model to be used based on familiarity, even when the method is suboptimal for the particular application (Camacho et al., 2009, 2012).
An alternative, more structured approach is to consider a set of candidate models and to select the best one on the basis of the performance on data not used in calibration (that is, in validation). An independent validation dataset can be leveraged for model selection and discrimination in the so-called hold-out validation (Bishop, 1995). If an independent validation dataset is not available or cannot be produced, model selection and discrimination can be achieved by using the calibration dataset alone and resampling techniques, cross-validation (Allen, 1974; Stone, 1974) being the most popular choice. Comparing a large number of models based on their performance in cross-validation is, in fact, the general principle underlying numerous frameworks for automated machine learning (AutoML; Hutter et al., 2019). Some notable software packages include Auto-sklearn (Feurer et al., 2015), AutoWEKA (Kotthoff et al., 2017), Auto-Keras (Jin et al., 2019), TPOT (Le et al., 2020), H2OAutoML (H2O AI, 2023), TransmogrifAI (Salesforce, 2021), and MLJAR (MLJAR, 2023).
However, all the available AutoML packages are designed to handle supervised learning problems (mostly regression), while no automated system is available to develop fault detection

systems. Furthermore, comparing a large number of candidate models by cross-validation has been proven to increase the chances of selecting a suboptimal model, especially when a limited amount of data is available (Arlot et al., 2010). A further issue is that the cross-validation procedure is generally the same for all the models being compared, and their characteristics are disregarded, which is particularly relevant if models able to cope with different characteristics of the data are compared. An example is the comparison of models for static or dynamic data: while standard cross-validation assumes that observations are independent (Arlot et al., 2010), special procedures are required for data featuring dynamics, in which observations are autocorrelated (Bergmeir et al., 2012). A final drawback of comparing multiple, possibly very different models based on cross-validation alone lays in the fact that such a "winner takes all" approach disregards the appropriateness of the chosen model to the characteristics of the data at hand, which are not considered at all. Therefore, an inappropriate and non-robust model could show the best performance by chance and still be selected.

The aforementioned limitations have been discussed and illustrated by (Sun et al., 2021). They proposed a bottom-up approach for automated model selection and calibration meant to tackle data-driven regression problems: smart process analytics (SPA). The procedure is based on a preliminary assessment of the relevant properties of the data at hand (correlation, nonlinearity, and dynamics), which allows to pre-select only appropriate models (meaning the ones that can cope with the detected characteristics) among the models included in the SPA library. A rigorous cross-validation approach tailored to the characteristics of the selected model category is then used to identify the best candidate. The most relevant difference between AutoML packages and SPA lays in the additional pre-selection step, based on the characteristics of the data at hand: it ensures that only appropriate models are compared by cross-validation, therefore effectively limiting the chances of overfitting.

In this Chapter, we propose a SPA-like approach for automatizing the selection and application of the best fault detection method for a given dataset: smart process analytics for process monitoring (SPAfPM). Section 7.2 provides an overview of fault detection methods included in the SPAfPM model library, their mathematical assumptions, and their characteristics. Section 7.3 describes the relevant characteristics sought after in the preliminary data interrogation step of SPAfPM. The design and assessment of the criteria employed for preliminary data interrogation is illustrated in Section 7.4. The model selection mechanism used in SPAfPM is introduced in Section 7.5. Finally, Section 7.6 demonstrates the effectiveness of SPAfPM on a variety of case studies, and conclusions are drawn in Section 7.7.

7.2 Data-driven methods for fault detection

The data-driven models included in SPAfPM are described in this Section. Each one of the models we introduce can cope with specific characteristics of the data, which are discussed in

Section 7.3. Fundamental, linear methods belonging to the family of latent-variable models are introduced first, with the statistics they provide to detect process faults and approaches to estimate their control limits. Nonlinear and dynamic models are considered as well.

7.2.1 Linear fault detection methods

Among the numerous methods available for fault detection, latent-variable models, such as PCA (Wold et al., 1987a) and PLS (Wold et al., 1987a, 2001), achieved a remarkable number of successful applications reported in a flourishing literature (Reis et al., 2017). CVA (Larimore, 1990) was used to tackle several fault detection problems as well (Chiang et al., 2001; Russell et al., 2000; Severson et al., 2016) by virtue of its ability to model process dynamics. These models, the rationales of which have been introduced in Sections 2.1, 2.2, and 2.3, represent the fundamental methods included in the SPAfPM model library. Their calibration requires to determine some hyperparameters: the hyperparameters of PCA and PLS are the numbers of PCs and LVs, respectively, while the hyperparameters of CVA are the memory order, the extent of the past horizon, and the extent of the future horizon. Once these hyperparameters are tuned, PCA, PLS, and CVA can be used to develop models of NOC data for fault detection, as outlined below.

PCA can be used for general process monitoring, meaning to detect faults affecting the process variables included in the modeled data matrix (Kourti et al., 1995, 1996; Nomikos et al., 1994, 1995a; Qin, 2003; Wise et al., 1996). The fault detection statistics of PCA, defined in Section 2.1.3, are:

- the T_X^2 statistic describes the variation of data within the space of PCs, hence it gives a measure of the distance of the state of the process from the NOC;
- the Q_X statistic describes the variation of data in the noise space, giving a measure of how much the state of the process strays form the correlation structure of the NOC.

PLS shares the same fault detection statistics of PCA, introduced in Section 2.2.4. However, as PLS models only the variance of input variables related to the variance of output variables, it finds applications in quality-relevant monitoring (Kourti et al., 1995, 1996; Nomikos et al., 1994, 1995a; Qin, 2003; Wise et al., 1996). In fact, the interpretation of statistics differs slightly with respect to PCA:

- the T_X^2 statistic describes the variation of data within the space of LVs, hence it gives a measure of the distance of the state of the process from the NOC in terms of input-output relationship;
- the Q_X statistic describes the variation of data in the noise space, giving a measure of how much the state of the process strays form the inputoutput cross-correlation structure of the NOC.

CVA provides similar statistics to PLS for quality-relevant monitoring, as outlined in Section 2.3.4. However, due to the ability of CVA to encode information on process dynamics and its

relationship to state-space modeling (Chiang et al., 2001; Larimore, 1990), the interpretation of its fault detection statistics is different from the ones of other models:

- the T_X^2 statistic describes variations inside the state-space, giving a measure of the distance of the process state from the NOC within the space of CVs in the main model;
- the Q_X statistic describes variations in the residual space, measuring the differences of the correlation and autocorrelation structures of the process from the ones of NOC data;
- the $T_{X,\mathrm{r}}^2$ statistic describes variations outside of the state-space, measuring the distance of the process state from the NOC within the space of residual CVs.

7.2.2 Control limits estimation approaches

Methods to estimate the confidence limits of the statistics discussed in the Section 7.2.1 have been mentioned as well in Sections 2.1.3, 2.2.4, and 2.3.4, referencing relevant literature resources. The methods of interest for SPAfPM are described in detail in this Section. Note that the wordings "confidence limit" and "control limit" share the same meaning in fault detection, the latter being a contextual interpretation of the former statistical concept.

The most common methods to estimate the control limits of the T_X^2 statistic are the ones based on the F distribution (Jackson, 1959) and on the χ^2 distribution with matching moments (Nomikos et al., 1995a). The control limit of T_X^2 at significance level α based on the F distribution is defined as:

$$T_{X,\mathrm{lim}}^2|_\alpha = \frac{\mathrm{DOF}(N-1)(N+1)}{N(N-\mathrm{DOF})} F_{1-\alpha}(\mathrm{DOF}, N-\mathrm{DOF}) \quad , \tag{7.1}$$

where DOF represents the degrees of freedom of the relevant model and $F_{1-\alpha}(\mathrm{DOF}, N-\mathrm{DOF})$ denotes the value of a F variable with DOF and $N-\mathrm{DOF}$ degrees of freedom at the numerator and denominator, respectively, evaluated at probability $1-\alpha$. The degrees of freedom to be used in (7.1) varies according with the model. However, the dominant approach in the literature is to use the number of PCs, LVs, and CVs for PCA, PLS, and CVA, respectively, thus $\mathrm{DOF} = A$. Note that this value is generally adopted also in the estimation of control limits of dynamic and nonlinear extensions of the basic models, which are described in the following. More sophisticated approaches for DOF estimation exist nonetheless for PCA (Hassani et al., 2012), PLS (Krämer et al., 2011; Van Der Voet, 1999), and CVA (Candy et al., 1979; Chiang et al., 2001; Ljung, 1999).The control limit of T_X^2 at significance level α based on the χ^2 distribution with matching moments is defined as:

$$T_{X,\mathrm{lim}}^2|_\alpha = \frac{s_{T_X^2}^2}{2\overline{T_X^2}} \chi_{1-\alpha}^2\left(2\left(\overline{T_X^2}/s_{T_X^2}\right)^2\right) \quad , \tag{7.2}$$

where $\overline{T_X^2}$ and $s_{T_X^2}$ are the sample mean and standard deviation of T_X^2, computed using the values of the statistic derived from the calibration dataset, while $\chi_{1-\alpha}^2\left(2\left(\overline{T_X^2}/s_{T_X^2}\right)^2\right)$ is the value of a χ^2 variable with $2\left(\overline{T_X^2}/s_{T_X^2}\right)^2$ degrees of freedom evaluated at probability $1-\alpha$. This equation can be applied to all the models described in this Section. Both the F-based and χ^2-based limits rely on the normality assumption of the values of T_X^2 obtained from the calibration dataset;

however, the χ^2 control limit is recommended in general by virtue of its mild robustness to violations of the normality assumption (Qin, 2003).

Common approaches for the determination of the control limits of the Q_X are the Jackson-Mudholkar method (Jackson et al., 1979) and the approach based on the χ^2 distribution with matching moments (Nomikos et al., 1995a). The Jackson-Mudholkar approach can be applied to PCA and PLS only (including their dynamic and nonlinear extensions), while it cannot be applied to CVA. It estimates the control limit of Q_X at significance level α as:

$$Q_{X,\text{lim}}|_\alpha = \theta_1 \left(1 + \frac{z(1-\alpha)h_0\sqrt{2\theta_2}}{\theta_1} + \frac{h_0(h_0-1)\theta_2}{\theta_1^2}\right)^{\frac{1}{h_0}} \quad , \tag{7.3}$$

where:

$$\theta_i = \sum_{j=A+1}^{V_X} \left(\lambda_j\right)^i \quad , \tag{7.4}$$

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \quad . \tag{7.5}$$

In (7.3), $z(1-\alpha)$ is the value of a standard normal variable evaluated at probability $1-\alpha$, while the λ_j in (7.4) are the variances of $V_X - A$ PCs/LVs not considered in the PCA/PLS model, which can be computed as in (2.10)/(2.24). Note that V_X in (7.4) must be replaced by the number of columns of the relevant input matrix when applying the Jackson-Mudholkar approach to extensions of PCA and PLS discusses in the following Sections. On the other hand, the control limit of Q_X at significance level α based on the χ^2 distribution with matching moments can be computed also for CVA and is defined as:

$$Q_{X,\text{lim}}|_\alpha = \frac{s_{Q_X}^2}{2\overline{Q_X}} \chi_{1-\alpha}^2 \left(2\left(\overline{Q_X}/s_{Q_X}\right)^2\right) \quad , \tag{7.6}$$

where $\overline{Q_X}$ and s_{Q_X} are the sample mean and standard deviation of the Q_X, computed using the values of the statistic derived from the calibration dataset, while $\chi_{1-\alpha}^2\left(2\left(\overline{Q_X}/s_{Q_X}\right)^2\right)$ is the value of a χ^2 variable with $2\left(\overline{Q_X}/s_{Q_X}\right)^2$ degrees of freedom evaluated at probability $1-\alpha$. This equation can be applied to all the models described in this Section. Similarly to the limits for T_X^2, both the Jackson-Mudholkar approach and the χ^2-based limits rely on the normality assumption of the values of Q_X obtained from the calibration dataset; the χ^2 control limit is recommended due to its robustness (Qin, 2003).

The CVA model provides an additional fault detection statistic: the $T_{X,\text{r}}^2$ statistic. Control limits for this statistic can be computed with the same approaches outlined for the T_X^2 statistic. The χ^2 distribution approach can be applied directly through (7.2), while the application of the F distribution approach in (7.1) requires to set $\text{DOF} = (V_X + V_Y)L - A$. Further details on the control limits can be found in the literature (Qin, 2003; Reis et al., 2021a; Thissen et al., 2001; Tracy et al., 1992).

All the approaches introduced so far rely on some assumptions on the control statistics, requiring compliance to a specific, static distribution. However, such assumptions could be violated in cases of high relevance for fault detection, for example when dynamic models are used (Ku et al., 1995). An alternative approach, free from any distributional assumption, relies

on non-parametric density estimation (Martin et al., 1996), for example kernel density estimation (KDE; Parzen, 1962; Rosenblatt, 1956).

KDE models the distributions of a given random variable X in terms of its probability density function (PDF). Given a sample $\mathbf{x} \in \mathbb{R}^N$ gathering observations x_n, with $n \in \{1, \dots, N\}$, of a random variable X, the PDF of X is estimated as:

$$p_s(s) = \frac{1}{N\delta} \sum_{n=1}^{N} k\left(\frac{x - x_n}{\delta}\right) \quad , \tag{7.7}$$

where x is a generic realization of X and k is a given kernel function, typically a radial basis function (RBF) kernel, also known as Gaussian kernel. The RBF kernel in defined as:

$$k\left(\frac{x - x_n}{\delta}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x - x_n)^2}{\delta^2}\right) \quad , \tag{7.8}$$

δ being the kernel width, a hyperparameter of the KDE estimator. Small values of δ yield a more irregular PDF, while the PDF converges to a wide Gaussian distribution as δ increases. Empirical rules have been proposed to determine the optimal width of the RBF kernel in KDE (Scott, 1992; Silverman, 1986). Two examples are the Scott rule:

$$\delta_{\text{opt}} = 1.059 s_{\mathbf{x}} N^{-\frac{1}{5}} \quad , \tag{7.9}$$

and the Silverman rule:

$$\delta_{\text{opt}} = 0.9 s_{\mathbf{x}} N^{-\frac{1}{5}} \quad . \tag{7.10}$$

In both cases, $s_{\mathbf{x}}$ is the sample standard deviation of X computed on the sample $\mathbf{x}$. Variants of both the rules have been proposed as well: in (7.9) and (7.10), $s_{\mathbf{x}}$ can be replaced by $\min\{s_{\mathbf{x}}, \text{IQR}/1.349\}$, where IQR represents the inter-quartile range of the sample $\mathbf{x}$.

The one-tail KDE-based confidence limit of the random variable X at a given significance level α can finally be defined as:

$$x_{\text{lim}}|_{\alpha} = x^* \text{ such that } \int_{-\infty}^{x^*} p_s(s)\, \mathrm{d}s = 1 - \alpha \quad . \tag{7.11}$$

KDE is a general method that can applied to any random variable, thus it can used to estimate the control limit of any fault detection statistic. In this Chapter, X can be T_X^2, Q_X, or $T_{X,\text{r}}^2$.

7.2.3 Dynamic transformations

Among the models mentioned in Section 7.2.1, only CVA is suitable to deal with dynamics in the dataset. The reason, briefly stated in Section 2.3.4, is that PCA and PLS respectively model covariance and cross-covariance matrices accounting only for static correlation among variable, thus leaving unmodeled the correlation among observations featured by data from dynamic processes (Bergmeir et al., 2012). This means that faults affecting process dynamics could be undetected by static methods (Ku et al., 1995). To account for dynamics, autocorrelation coefficients (Box et al., 2016) and cross-correlation coefficients (Brockwell et al., 2016) can be considered in modeling by augmenting the data matrices by means of lagged measurements prior to model calibration (Ku et al., 1995). In fact, a number of dynamic extensions of the basic methods have been proposed, among which dynamic principal component analysis (DPCA; Ku et al., 1995) and dynamic partial least-squares (DPLS; Ricker, 1988) regression.

DPCA aims at modeling autocorrelation and cross-correlation in the dataset, implicitly extracting a dynamic autoregressive model of the process (Ku et al., 1995). In this case, the rows of the data matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$ are assumed to be observations of the realization of a multivariable random process at a sequence of times, hence a given row depends on previous ones. To include the correlation between the different time instants (observations), an augmented matrix is formed by lagged measurement augmentation as:

$$\boldsymbol{\mathcal{X}}_L = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} & \mathbf{x}_2^{\mathrm{T}} & \cdots & \mathbf{x}_{L+1}^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} & \mathbf{x}_3^{\mathrm{T}} & \cdots & \mathbf{x}_{L+2}^{\mathrm{T}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{N-L}^{\mathrm{T}} & \mathbf{x}_{N-L+1}^{\mathrm{T}} & \cdots & \mathbf{x}_N^{\mathrm{T}} \end{bmatrix} , \tag{7.12}$$

where $\mathbf{x}_n \in \mathbb{R}^{V_X}$ is the n-th row (observation) of $\mathbf{X}$. The matrix $\boldsymbol{\mathcal{X}}_L$ obtained in such a way includes the L lagged measurements as additional variables, besides the measurement at the current time instant (last block of V_X columns), which results in $V_X(L+1)$ columns. For rows of $\boldsymbol{\mathcal{X}}_L$ to be complete, the first observation refers to time $L+1$, resulting in $N-L$ rows. Therefore, $\boldsymbol{\mathcal{X}}_L \in \mathbb{R}^{N-L} \times \mathbb{R}^{V_X(L+1)}$; note that $\boldsymbol{\mathcal{X}}_0 = \mathbf{X}$. Usually one or two lags are regarded to be sufficient to represent most of the relevant dynamics (Kini et al., 2019; Lee et al., 2004b).

DPCA is performed by applying the conventional PCA described in Section 2.1 to matrix $\boldsymbol{\mathcal{X}}_L$. The general PCA equations for T_X^2 and Q_X still hold true. However, dealing with dynamic data requires additional considerations. In the context of regression, using a static regression model leaves dynamic structures in the regression residuals (Sun et al., 2021). A similar phenomenon can be observed for PCA: using a static version of PCA results in dynamic effects in the Q_X statistic; similarly, using a dynamic version of PCA, such as DPCA incorporates dynamic effects into the model, resulting in dynamic scores and correlated T_X^2 statistic (Ku et al., 1995). However, the control limit based on the F distribution assumes an uncorrelated T_X^2 statistic. Consequently, the χ^2-based control limit is recommended when applying dynamic extensions of latent-variable models, such as DPCA (Lu et al., 2005; Yao et al., 2007).

Dynamic extensions of PLS can be constructed in a similar way as for DPCA. PLS uses an input data matrix $\mathbf{X}$ and an output data matrix $\mathbf{Y}$, and both of them could be augmented by lagged measurements. However, in the most common approach to DPLS, only $\mathbf{X}$ is augmented, while $\mathbf{Y}$ is left unchanged. Therefore, DPLS is obtained by modeling the lag-augmented input matrix $\boldsymbol{\mathcal{X}}_L$ defined in (7.12) and the output matrix $\mathbf{Y}$ by the standard PLS introduced in Section 2.2. This version of DPLS yields a finite input response representation of the dynamics (Jia et al., 2016; Jiao et al., 2015; Ricker, 1988). It is recommended to use the χ^2-based control limit for T_X^2 for the same reason as for DPCA.

Besides to CVA, DPCA and DPLS are included in SPAfPM as fault detection models to deal with dynamics in the data. Compared to standard PCA and PLS, only the number of lags is to be determined as an additional hyperparameter. However, it is worth mentioning that other dynamic latent-variable models exist, including DPCA with decorrelated residuals (Rato et al., 2013), dynamic-inner PCA (Dong et al., 2020, 2018a, 2021), dynamic-inner PLS (Dong et al.,

2015, 2018b), and auto-regressive PLS (Zhu, 2021). These models differ in how dynamics is represented. See Fernandes et al. (2022) for details.

7.2.4 Nonlinear transformations

PCA and PLS are linear modeling methods. However, real-world data often feature some degree of nonlinearity. Several nonlinear extensions of the aforementioned methods have been proposed in the literature. Nonlinear versions of PCA can be obtained in various ways, for examples: by applying nonlinear transformations to the data matrix, as in kernel principal component analysis (KPCA; Schölkopf et al., 1998); by means of generalized algorithms aimed at identifying nonlinear principal components, an example being principal curves (Hastie et al., 1989); representing PCA as a neural network structure, which yields neural network principal component analysis (NNPCA) in different forms, such as auto-associative neural networks (Kramer, 1991), input-training networks (Tan et al., 1995), and double-network strategies (Dong et al., 1996). Differently from PCA, PLS can be made nonlinear by two main strategies (Rosipal, 2010):

- replacing the inner linear regression model with a nonlinear one;
- applying nonlinear transformations to the **X** matrix (possibly also to the **Y** matrix) and using the transformed matrix as input to linear PLS model.

Concerning the first category, the linear regression model of PLS can be replaced by a quadratic regression model (Wold et al., 1989), a spline regression model (Wold, 1992), or by neural network models in various configurations (Malthouse et al., 1997; Qin et al., 1992; Wilson et al., 1997). Regarding the second category, kernel partial least squares regression (KPLS; Rosipal et al., 2001) relies on the same rationale of KPCA: a standard PLS model is calibrated using a kernel-transformed input matrix.

Models based on kernel transformations (KPCA and KPLS) are selected for inclusions into the SPAfPM model library due to their computational efficiency and proven effectiveness. Approaches based on neural networks were discarded due to their several drawbacks. For example, considering NNPCA, tuning the structures of the relevant networks is regarded to be a very challenging task (Tan et al., 1995), and no guarantee is given on the actual orthogonality of the nonlinear PCs obtained in such a way (Jia et al., 2000). This in turn causes issues in the definition of the T_X^2 and Q_X statistics for fault detection (Thissen et al., 2001). Furthermore, neural network-based PCA variants are known to be plagued by the issue of local minima and to entail a high computational workload for calibration (Sun, 2020a). Similar drawbacks affect extension of PLS based on neural networks. However, PLS extensions based on nonlinear inner regression models represent a promising future research path (Rosipal, 2010), also for nonlinear quality-relevant fault detection.

KPCA represents and effective way to include nonlinearity into the basic PCA model (Schölkopf et al., 1998). Given an observation $\mathbf{x} \in \mathbb{R}^{V_X}$ of the V_X input variables, assumed to

be nonlinearly correlated, a mapping function $\phi : \mathbb{R}^{V_X} \to \mathbb{R}^E \mid \mathbf{x} \mapsto \phi(\mathbf{x})$ projects the input variables onto a high-dimensional space, commonly referred to as the feature space; E is the dimension of the feature space and can be arbitrarily large. Such mapping function is defined as to capture nonlinear relationships in the original data, effectively linearizing the correlation among variables in the feature space. The covariance matrix of a data matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$ (the columns of which are assumed to be at least mean-centered) can be computed as:

$$\mathbf{S}_X = \frac{1}{\mathrm{N}-1}\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X} \quad . \tag{7.13}$$

While the PCA calibration procedure introduced in Section 2.1.1 relies on the direct SVD of the data matrix $\mathbf{X}$, an equivalent formulation can be given based on the SVD of the covariance matrix $\mathbf{S}_X$. An alternative formulation of the covariance matrix in (7.13) based on the observations in $\mathbf{X}$, denoted as $\mathbf{x}_n \in \mathbb{R}^{V_X}$, is:

$$\mathbf{S}_X = \frac{1}{\mathrm{N}-1}\sum_{n=1}^{N} \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_n \quad . \tag{7.14}$$

The latter formulation allows to define the covariance matrix of the input variables transformed by the mapping function ϕ:

$$\mathbf{S}_{\phi(X)} = \frac{1}{\mathrm{N}-1}\sum_{n=1}^{N} \phi(\mathbf{x}_n^{\mathrm{T}}) \cdot \phi(\mathbf{x}_n) \quad . \tag{7.15}$$

SVD can then be used to model the transformed covariance matrix, as in regular PCA.

A drawback of this strategy is that an explicit mapping of the observations could cause the procedure to be computationally unfeasible due to the arbitrarily large dimension of the feature space; however, an explicit mapping by the function ϕ is not needed in practice, as the kernel trick (Schölkopf et al., 1998) allows to implicitly compute dot (scalar) products between vectors in the feature space, which is all that is required to solve the singular value decomposition (note that the covariance matrix in (7.15) is defined uniquely in terms of dot products). First, a pairwise kernel function complying with the conditions of Mercer's theorem (Müller et al., 2001; Schölkopf et al., 1999) is be defined as:

$$k_{\mathrm{p}} : \mathbb{R}^{V_X} \times \mathbb{R}^{V_X} \to \mathbb{R} \mid [\mathbf{x}_n \quad \mathbf{x}_m] \mapsto \phi(\mathbf{x}_n^{\mathrm{T}}) \cdot \phi(\mathbf{x}_m) \quad . \tag{7.16}$$

Note that this function is different from the kernel used in KDE (see Section 7.2.2), as k_{p} is applied to pairs of observations and returns a scalar. Several pairwise kernel functions exist. Among the most common kernels are the RBF kernel, also known as Gaussian kernel:

$$k_{\mathrm{rbf}}(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2}{\sigma^2}\right) \quad , \tag{7.17}$$

where $\|\cdot\|$ is the Euclidean norm and σ is the kernel bandwidth, and the polynomial kernel:

$$k_{\mathrm{poly}}(\mathbf{x}_n, \mathbf{x}_m) = (\gamma(\mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_m) + c_0)^d \quad , \tag{7.18}$$

where γ is again know as kernel bandwidth, while c_0 and d are the polynomial offset and degree, respectively. With reference to fault detection problems, as the bandwidth becomes larger for the RBF kernel and smaller for the polynomial kernel, model robustness increases whereas model sensitivity decreases (Choi et al., 2004).

This pairwise kernel function is applied to all the possible couples of observations in the input data matrix $\mathbf{X}$ to obtain the components of a kernel matrix $\mathbf{K} \in \mathbb{R}^N \times \mathbb{R}^N$, the entries of which

represent the dot products of couples of transformed observations implicitly computed in the feature space. The component $K_{n,m}$ of the kernel matrix is defined as:

$$K_{n,m} = k_{\mathrm{p}}(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n^{\mathrm{T}}) \cdot \phi(\mathbf{x}_m) \quad . \tag{7.19}$$

The kernel matrix must be pre-processed by mean-centering prior to modeling. However, as observations are implicitly mapped to the feature space, mean-centering must be performed therein as (Schölkopf et al., 1998):

$$\widetilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N \cdot \mathbf{K} - \mathbf{K} \cdot \mathbf{1}_N + \mathbf{1}_N \cdot \mathbf{K} \cdot \mathbf{1}_N \quad , \tag{7.20}$$

where $\mathbf{1}_N \in \mathbb{R}^N \times \mathbb{R}^N$ is a matrix containing components all equal to N^{-1}. The number of PCs to be extracted in the feature space is set to A and, said $R = \mathrm{rank}(\widetilde{\mathbf{K}})$, the centered kernel matrix $\widetilde{\mathbf{K}}$ is decomposed by SVD:

$$\widetilde{\mathbf{K}} = \mathbf{N} \cdot \mathbf{\Sigma} \cdot \mathbf{O}^{\mathrm{T}} = [\mathbf{N}_1 \quad \mathbf{N}_2] \cdot \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \cdot [\mathbf{O}_1 \quad \mathbf{O}_2]^{\mathrm{T}} \quad , \tag{7.21}$$

where the dimensions of the matrices involved in the SVD are similar to the ones mentioned in Section 2.1.1. Finally, the loadings of the KPCA model, $\mathbf{P} \in \mathbb{R}^N \times \mathbb{R}^A$, are:

$$\mathbf{P} = \mathbf{\Sigma}_1^{-1} \cdot \mathbf{O}_1 \quad , \tag{7.22}$$

where the normalization of the singular vectors by the singular values is required to ensure that the loadings are normalized to unit norm in the feature space. The scores of the input observation in the feature space, $\mathbf{T} \in \mathbb{R}^N \times \mathbb{R}^A$, can then be computed as:

$$\mathbf{T} = \widetilde{\mathbf{K}} \cdot \mathbf{P} \quad . \tag{7.23}$$

The pairwise kernel function used to compute entries of $\mathbf{K}$, including the hyperparameters of the said function, must be set prior to KPCA modeling, therefore they are hyperparameters for KPCA (together with the number of PCs) considered in SPAfPM. Despite the large number of possible combinations, the direct calibration algorithm makes KPCA still computationally more efficient than NNPCA, even when using an extensive cross-validation procedure.

Given a new observation $\mathbf{x}_{\mathrm{new}} \in \mathbb{R}^{Vx}$, the KPCA model can be used to compute its scores in the feature space. First, a new kernel vector $\mathbf{k}_{\mathrm{new}} \in \mathbb{R}^N$ is computed component by component as:

$$k_{\mathrm{new}_n} = k_{\mathrm{p}}(\mathbf{x}_n, \mathbf{x}_{\mathrm{new}}) = \phi(\mathbf{x}_n^{\mathrm{T}}) \cdot \phi(\mathbf{x}_{\mathrm{new}}) \quad , \tag{7.24}$$

where $\mathbf{x}_n$ is the n-th observation in the calibration matrix $\mathbf{X}$. The kernel vector is then pre-processed as to center it in the feature space using the pre-processing parameters of the calibration data:

$$\tilde{\mathbf{k}}_{\mathrm{new}}^{\mathrm{T}} = \mathbf{k}_{\mathrm{new}}^{\mathrm{T}} - \mathbf{1}'_N \cdot \mathbf{K} - \mathbf{k}_{\mathrm{new}}^{\mathbf{T}} \cdot \mathbf{1}_N + \mathbf{1}'_N \cdot \mathbf{K} \cdot \mathbf{1}_N \quad , \tag{7.25}$$

where $\mathbf{1}'_N \in \mathbb{R} \times \mathbb{R}^N$ is a matrix containing components all equal to N^{-1}. Finally, the scores of the new observation in the feature space are computed as:

$$\mathbf{t}_{\mathrm{new}}^{\mathrm{T}} = \tilde{\mathbf{k}}_{\mathrm{new}}^{\mathrm{T}} \cdot \mathbf{P} \quad . \tag{7.26}$$

KPLS is based on the same idea as KPCA. In the most common version of KPLS (Rosipal et al., 2001), the $\mathbf{X}$ matrix is transformed by means of a pairwise kernel function, while the $\mathbf{Y}$ matrix is left unchanged (similarly to the DPLS introduced in Section 7.2.3). The conventional

PLS calibration algorithm described in Section 2.2.1 is then applied to the mean-centered kernel matrix $\widetilde{\mathbf{K}}$ and to the autoscaled $\mathbf{Y}$ matrix. An iterative procedure is required to extract sequentially the A LVs. The reader is referred to literature resources for details on the KPLS calibration procedure (Jia et al., 2016; Rosipal et al., 2001; Wang et al., 2014). Note that KPLS requires the identification of the same hyperparameters as KPCA.

The T_X^2 and Q_X statistics can be computed for both KPCA and KPLS using the same equations valid for PCA and PLS, respectively (see Sections 2.1.3 and 2.2.4). Note that, as the T_X^2 statistic is computed in the feature space, therefore considering the kernel-transformed NOC data, its distribution is expected to be normal (Choi et al., 2004), which enables the estimate the control limits by approaches based on the F and χ^2 distributions. In fact, the goal of kernel methods is to map data from a general distribution in the input space to a normal distribution in the feature space. The validity of this assumption is shown in Section 7.3.2. Similarly, the Q_X statistic can be computed as the difference between the mean-centered kernel matrix and the reconstruction of the same matrix yielded by the relevant model, as in PCA and PLS. The Jackson-Mudholkar approach or the χ^2 method can be used to estimate the control limits of Q_X (Cho et al., 2005; Choi et al., 2005; Zhang et al., 2008).

7.2.5 Combination of dynamics and nonlinearity

Due to its relationship to SVD, CVA can model linear dynamics in the data. The same holds true for DPCA and DPLS, as they rely on autocorrelation coefficients (Ljung, 1999). While a tight control system may effectively linearize the dynamics and correlation in continuous processes (Sun, 2020a), nonlinear dynamics is a common occurrence in data from real manufacturing processes. Extensions of the dynamic and nonlinear methods outlined in the previous Sections have been proposed to handle combinations of nonlinearity and dynamics, for example by dynamic kernel principal component analysis (DKPCA; Choi et al., 2004) and dynamic kernel partial least-squares (DKPLS; Jia et al., 2016) regression. In both these methods, the input matrix $\mathbf{X}$ is first augmented by L lagged measurements to obtain matrix $\boldsymbol{\mathcal{X}}_L$, as in (7.12), then this matrix is processed by kernel transformation as in (7.19). Therefore, DKPCA combines DPCA and KPCA, while DKPLS combines DPLS and KPLS (the $\mathbf{Y}$ matrix is left unchanged in this case).

While such a simple combination may seem naïve, it can effectively handle both nonlinearity and dynamics in the data (Choi et al., 2004; Jia et al., 2016). Several literature studies support this point. For example, Baffi et al. (2000) argue that nonlinear dynamics can be modeled using nonlinear PLS applied to lag-augmented matrices, and Choi et al. (2004) argue the same in the context of nonlinear PCA. An intuitive understanding is provided by the following reasoning. Augmenting data matrices by lagged measurements allows to consider correlation between observations (in the form of autocorrelation and cross-correlation) when computing the covariance matrices modeled by DPCA and DPLS. However, being these models based on

conventional PCA and PLS, they can extract linear correlation only, hence only linear dynamic is modeled, and any nonlinear dynamic in the data is left unmodeled as not fully represented by autocorrelation and cross-correlation coefficients (this point is further discussed in Section 7.3.2 for the case of static correlation). On the other hand, applying nonlinear (kernel) transformations to the lag-augmented matrices prior to modeling linearizes the nonlinear correlation among observations in the feature space, which is therefore available to modeling therein. The combination of augmentation by lagged measurement (to include dynamic information) and of kernel transformation (to linearize such dynamic information) can therefore effectively model nonlinear dynamics in the data. In fact, this concept shares remarkable similarities with the Hammerstein model philosophy (Ljung, 1999), which was successfully combined with latent-variable modeling to describe systems featuring nonlinear dynamics (Lakshminarayanan et al., 1995, 1997).

Given the strong justification just outlined and their computational simplicity, DKPCA and DKPLS are included in the SPAfPM model library for fault detection on processes featuring nonlinear dynamics. Fault-detection statistics and control limits can be formulated in the same way as for KPCA and KPLS (Choi et al., 2004; Jia et al., 2016). See Section 7.2.4 for details. The hyperparameters to be determined for both models are the number of PCs/LVs, the number of lags, and the kernel function with its hyperparameters.

Several extensions of CVA to nonlinear systems have been proposed as well. Examples are: methods based on basis expansion by nonlinear transformations of input and output variables prior to lag-augmentation (Lakshminarayanan et al., 1995); methods based on general nonlinear transformations of the past and future matrices (Larimore et al., 1990); methods based on transformations of the original input matrix **X** prior to standard CVA modeling, for example by KPCA (Samuel et al., 2015a, 2015b); combination of standard CVA with KDE (Odiowei et al., 2009, 2010). The latter approach, referred to as KDE-CVA in this Chapter, is particularly appealing due to its architectural simplicity and its proven effectiveness in capturing nonlinear effects in the monitoring statistics when applied to data featuring nonlinear dynamics. This is achieved by a two-step procedure: a CVA model is first developed on the data at hand; then, KDE (see Section 7.2.2) is applied to the samples of the three monitoring statistics of CVA (T_X^2, Q_X, and $T_{X,\mathrm{r}}^2$) obtained in calibration to compute their PDFs and, therefore, their control limits. KDE-CVA with Gaussian kernel is included in the SPAfPM model library. With respect to standard CVA, only the kernel widths for the three fault detection statistics are to be identified as additional hyperparameters. However, KDE is implemented in SPAfPM as:

$$p_s(s) = \frac{1}{N\xi_s\delta_{\mathrm{opt}_s}} \sum_{n=1}^{N} k\left(\frac{x-x_n}{\delta_{\mathrm{opt}}}\right) \qquad s \in \left\{T_X^2, Q_X, T_{X,\mathrm{r}}^2\right\} \quad , \tag{7.27}$$

where the optimal kernel width for statistic s, δ_{opt_s}, is determined by the Scott rule in (7.9), and ξ_s is a scaling factor for the optimal kernel width. Therefore $\xi_{T_X^2}$, ξ_{Q_X}, and $\xi_{T_{X,\mathrm{r}}^2}$ are the additional hyperparameters identified by SPAfPM for KDE-CVA (compared to linear CVA).

7.2.6 Support vector data description

An alternative way to tackle the fault detection problem is to develop a one-class classification (OCC; Brereton, 2011) model of the NOC data. OCC methods aim to construct a description of data coming from a single class and to determine whether a new observation conforms to the characteristic of the modeled class or not (Rodionova et al., 2016). Therefore, OCC methods can be used to detect observations that significantly differ from the modeled class (Tax et al., 1999), distributional outliers (Tax et al., 2004), new data conditions (Rodionova et al., 2016), or to solve classification problems where one class is severely undersampled or missing overall. OCC is particularly useful in real-world fault detection problems, where it could be very expensive or not possible at all to generate faulty samples and to conjecture every possible fault (Tax et al., 2004). The OCC procedure is in fact more appropriate when no *a priori* assumption can be done about the distribution of the out-of-class (faulty) data (Tax et al., 1999).

An OCC model can be calibrated by estimating the PDF of the data in the modeled class, which gives a probabilistic view of the fault detection problem. However, density estimation is known to be a hard task (Müller et al., 2001), especially with high-dimensional data, due to the inherent sparsity of samples of multivariate distributions (Mecklin et al., 2005). Furthermore, the dataset could be biased if more NOC regions exist and some are more frequent than others, which could lead the model to focus on high density areas and to reject low density regions, though still belonging to NOC data (Tax et al., 2004). Estimating the support of the distribution of the modeled class is often enough (Müller et al., 2001), which implies modeling only the boundaries of the class.

Kernel transformations can be combined with OCC to solve fault detection problems concerning non-normal/nonlinear data due to the relationship between nonlinear mapping functions and implicit scalar product computation in the feature space performed by means of kernel functions (Müller et al., 2001). One method taking advantage of this synergy is support vector data description (SVDD; Tax et al., 1999, 2004).

For ease of understanding, the linear version of SVDD is described first. Given a data matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^{V_X}$ containing N observations of V_X variables, SVDD finds the hypersphere with the smallest radius that encloses all observations in the calibration matrix $\mathbf{X}$. The radius can be strongly influenced by outliers in the data; therefore, some observations are allowed to lay outside the hypersphere, trading domain coverage and rate of "misclassification". This allowance is achieved by means of N slack variables, $\zeta_n \in [0, +\infty)$, $n \in \{1, \dots, N\}$, and a radius-to-coverage parameter, $C \in [N^{-1}, 1]$. The radius of the hypersphere, $\mathcal{R}$, is identified by solving the optimization problem:

$$\begin{aligned} &(\mathcal{R}, \mathbf{a}) = \underset{\mathcal{R},\mathbf{a}}{\operatorname{argmin}}[\mathcal{R}^2 + C \textstyle\sum_{n=1}^{N} \zeta_n] \\ &\text{s.t. } \|\mathbf{x}_n - \mathbf{a}\| \leq \mathcal{R}^2 + \zeta_n, \quad n \in \{1, \dots, N\} \end{aligned} \quad , \tag{7.28}$$

where $\mathbf{x}_n \in \mathbb{R}^{V_X}$ is the n-th observation in $\mathbf{X}$ and $\mathbf{a} \in \mathbb{R}^{V_X}$ is the center of the hypersphere.

The center of the hypersphere is defined as a linear combination of observations:

$$\mathbf{a} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n \quad , \tag{7.29}$$

which can be demonstrated by incorporating the constraint to the optimization problem in (7.28) into the objective function by means of the Lagrange multipliers α_n, $n \in \{1, \dots, N\}$ (Tax et al., 2004). The multipliers can be determined solving of the aforementioned optimization problem and are themselves subject to two constraints:

$$0 \leq \alpha_n \leq C, \quad n \in \{1, \dots, N\} \quad , \tag{7.30}$$

$$\sum_{n=1}^{N} \alpha_n = 1 \quad . \tag{7.31}$$

Only observations $\mathbf{x}_n$ with non-zero multipliers are needed to describe the center of the hypersphere and are known as support vectors. Observations such that $\alpha_n \in (0, C)$ and $\zeta_n = 0$ lie on the edge of the hypersphere and can be used to compute its radius, while observations such that $\alpha_n = C$ and $\zeta_n > 0$ lie outside of the hypersphere. Given any support vector $\mathbf{x}_k \in \mathbb{R}^{Vx}$, the radius of the hypersphere is given by:

$$\mathcal{R}^2 = \mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{x}_k - 2\sum_{n=1}^{N} \alpha_n \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_k + \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_m \alpha_m \quad . \tag{7.32}$$

Furthermore, (7.32) can be used to compute the distance of a new observation $\mathbf{x}_{\mathrm{new}} \in \mathbb{R}^{Vx}$ from the center of the hypersphere:

$$D_{\mathrm{new}} = \sqrt{\mathbf{x}_{\mathrm{new}}^{\mathrm{T}} \cdot \mathbf{x}_{\mathrm{new}} - 2\sum_{n=1}^{N} \alpha_n \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_{\mathrm{new}} + \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_m \alpha_m} \, , \tag{7.33}$$

which allows to classify $\mathbf{x}_{\mathrm{new}}$ as within the modeled class or not (meaning conforming to the NOC or faulty, in fault detection applications). The condition to be satisfied for $\mathbf{x}_{\mathrm{new}}$ to be within the hypersphere is:

$$\mathbf{x}_{\mathrm{new}}^{\mathrm{T}} \cdot \mathbf{x}_{\mathrm{new}} - 2\sum_{n=1}^{N} \alpha_n \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_{\mathrm{new}} + \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \mathbf{x}_n^{\mathrm{T}} \cdot \mathbf{x}_m \alpha_m \leq \mathcal{R}^2 \quad . \tag{7.34}$$

As for the linear form of SVDD, a hypersphere can model boundaries of random vectors following an isotropic normal distribution. However, the objective function of SVDD can be written solely in terms of scalar products between observations in the input space (Tax et al., 1999, 2004). Furthermore, the equations to compute the radius of the hypersphere, to compute the distance of a new observation, and to test a new observation, that are (7.32), (7.33), and (7.34), respectively, are naturally written in terms of scalar products only. Therefore, SVDD can be made flexible (nonlinear) by replacing all the scalar products with the results of pairwise kernel functions as to implicitly compute those scalar products in a high-dimensional feature space, onto which observations of the input space are implicitly projected by nonlinear mapping functions. For example, the equation of the radius of the hypersphere in the feature space is:

$$\mathcal{R}^2 = k_{\mathrm{p}}(\mathbf{x}_k, \mathbf{x}_k) - 2\sum_{n=1}^{N} \alpha_n k_{\mathrm{p}}(\mathbf{x}_n, \mathbf{x}_k) + \\ + \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m k_{\mathrm{p}}(\mathbf{x}_n, \mathbf{x}_m) \quad , \tag{7.35}$$

and the condition for a new observation to fall within the hypersphere is:

$$k_{\mathrm{p}}(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}}) - 2\sum_{n=1}^{N} \alpha_n k_{\mathrm{p}}(\mathbf{x}_n, \mathbf{x}_{\mathrm{new}}) + \\ + \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m k_{\mathrm{p}}(\mathbf{x}_n, \mathbf{x}_m) \leq \mathcal{R}^2 \quad . \tag{7.36}$$

In principle, any pairwise kernel function as defined in (7.16), such as the polynomial kernel and Gaussian kernel introduced in Section 7.2.4, can be used in nonlinear SVDD. However, it

is known that the polynomial kernel can yield unsatisfactory results in some cases due to "extreme" observations dominating the inner product in (7.18) when the polynomial degree increases; for an example pf such behavior, see Tax et al. (1999). On the other hand, the Gaussian kernel does not suffer from such a drawback as it does not depend on the absolute position of observations, but just on their relative positions (Tax et al., 2004). Nonlinear SVDD with Gaussian kernel is also closely related to KDE (Tax et al., 2004), which further backs up its ability to solve OCC problems on general non-normal/nonlinear data. Furthermore, the properties of the Gaussian kernel allow to control the false alarm rate in fault detection problems, namely by tuning the kernel width (Tax et al., 2004).
Nonlinear SVDD is included in the SPAfPM model library as the only method not belonging to the family of latent-variable models. The hyperparameters to be determined for SVDD are the pairwise kernel function (including its relevant hyperparameters) and the radius-to-coverage parameter. Furthermore, SVDD offers important advantages when the data at hand include discrete variables. This point is further discussed in Section 7.3.3.

7.3 Data characteristics relevant to fault detection

The first step in the SPAfPM framework is a preliminary interrogation of data to infer characteristics relevant to fault detection problems. Such characteristics are used to perform a preliminary screening of the models provided with SPAfPM and are describe in this Section.

7.3.1 Data analytics triangle of SPAfPM

The performance of each model reviewed in Section 7.2 can vary significantly when applied to different datasets due to the underlying assumptions of the methods, which could or could not match to the characteristics of the data. Similarly to other smart data analytics approaches (Mohr et al., 2019; Sun et al., 2021), a base method can be identified and its assumptions used to determine the characteristics to be searched for in the available data. The characteristics found in a given dataset guide SPAfPM to the choice of the best model for the data at hand.
The base method chosen for the proposed framework is PCA by virtue its wide usage in the process monitoring literature and proven performance in fault detection. PCA can cope by design with datasets including a large numbers of possibly correlated variables, a feature that is reasonable to expect in data from manufacturing processes (Wise et al., 1996), often including measurements of all available process variables. PCA relies on three assumptions:

- correlation among variables is linear (Camacho et al., 2008a; Wold et al., 1987a);
- data follow a multivariate normal distribution (Qin, 2003);
- no dynamics is found in the data and/or residuals (Ku et al., 1995).

The assumptions of linear correlation and absence of dynamics are required due to the PCA working principle that defines latent variables as static, linear combinations of input variables

(see Section 2.1.1). On the other hand, the normality assumption is required to guarantee the reliability of the monitoring statistics and of their control limits. In fact, the matrix decomposition of PCA is based on the covariance matrix of data, encoding second-order information that can describe exactly only centered multivariate normal distributions. Furthermore, the control limits of the monitoring statistics discussed in Section 7.2.2, namely the ones for T_X^2 and Q_X, are fully descriptive only under the assumption that scores and residuals are normally distributed (Thissen et al., 2001). The scores are normally distributed only if the input data are normally distributed, as a linear combination of normal variables is still normal (Nomikos et al., 1994). On the other hand, residuals are normally distributed only if all the systematic variability (including the potential dynamics) is captured by the model and transferred to the latent space (Wold et al., 1987a).

As argued in Section 7.3.2, non-normality and nonlinearity are tightly intertwined characteristics of a dataset. Furthermore, methods able to cope with nonlinear correlation based on kernel transformations, such as KPCA and SVDD, can deal by default with general distributions (recall the discussion at the end of Section 7.2.6). Therefore, non-normality and nonlinearity are checked independently in SPAfPM, but are considered as a single characteristic of data. On the other hand, the presence of dynamics in the data is another characteristic to be assessed, therefore it is checked separately.

The objective of fault detection is a relevant information in SPAfPM: whether the monitoring system should detect all process faults regardless of their effects on the product quality, or only faults affecting the product quality. In other words, whether to adopt a "general" monitoring scheme or a quality-relevant monitoring approach (Li et al., 2011; Nomikos et al., 1995b). While PCA can be used to monitor the overall process, quality-relevant monitoring requires to model the process-product quality correlation. Provided that online measurements of the product quality variables are available, this can be achieved, for example, by PLS, which relies on assumptions similar to the ones of PCA. Therefore, a third relevant characteristic of the data is the availability of online variables describing the product quality (also referred to as dependent variables in this Chapter). However, the designation of quality variables requires expert knowledge on the process and product. Consequently, whether such variables are available or not is a choice left to the user and it is not automatically performed by SPAfPM.

To summarize, the core data characteristics considered in SPAfPM are:

- nonlinear correlation among variables/non-normal distribution of data;
- dynamics in the data (autocorrelation among observations);
- availability of dependent variables to describe the product quality.

These data characteristics and the data analytics methods included in the proposed framework and able to cope with such characteristics, which have been widely discussed in Section 7.2, can be visualized in the form of a smart data analytics triangle for fault detection: this is reported in Figure 7.2.

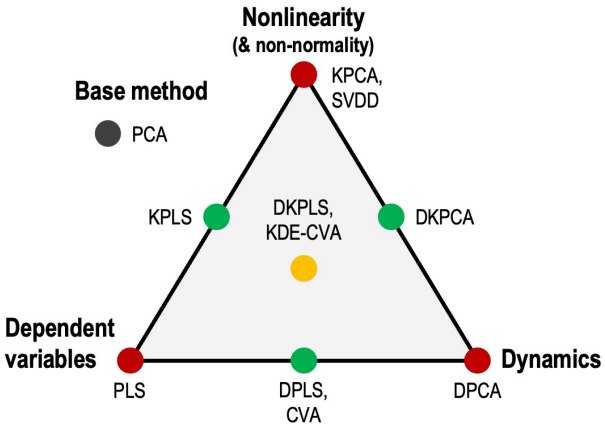


Figure 7.2. *The smart data analytics triangle for fault detection.*

The data analytics triangle in Figure 7.2 represents the model pre-selection operated by SPAfPM after the preliminary data interrogation. The corners represent models able to cope with one of the characteristics. The edges show the fault detection models suitable for the characteristics at the linked corners. The center of the triangle shows models best suited if all three characteristics are present in the dataset. If none of the relevant data characteristics is detected, the base method (PCA, outside of the triangle) is chosen. The criteria used to assess the presence of the characteristics in the triangle are discussed in Section 7.4.

As a matter of example, if the data feature nonlinearity and the presence of dependent variables is notified by the user, the data triangle suggests KPLS. Only one method is suggested in this case. However, if the data feature dynamics and dependent variables are present, two different models are recommended: DPLS and CVA. In this case, a model selection procedure is employed to determine which of the two methods is best for the given case (and to determine the optimal hyperparameters of the chosen model). An overview of the different hyperparameters for each one of the models in the data analytics triangle is shown in Table 7.1. The structure of the model selection procedure is explained in detail in Section 7.5.

7.3.2 Relationship between non-normality and nonlinearity

If the data are normally distributed, they feature linear correlation only. This point is intuitively supported by the following reasoning. First, the PDF of the multivariate normal distribution can be completely determined given the first two moments, namely the mean vector and the covariance matrix (Izenman, 2008). Then, the covariance matrix conveys the same information of the correlation matrix, the latter being a scaled version of the former. Finally, the correlation matrix measures exclusively the linear relationships between couples of variables, while it might not be sensitive to any nonlinear relationship among variables (Montgomery et al., 2018). The last point supports an intuitive understanding that, if variables feature nonlinear correlation, then their distribution cannot be normal. This fact is further backed up by two necessary conditions for a distribution to be multivariate normal: the marginal distributions of all variables

Table 7.1. *Overview of the hyperparameters of each model considered in SPAfPM.*

Model	Hyperparameter	Definition	Constraint
PCA	A	Number of PCs	
PLS	A	Number of LVs	
CVA	L	Extent of past horizon	
	H	Extent of future horizon	$H = L$
	A	Memory order	
DPCA	L	Number of lags	
	A	Number of PCs	
DPLS	L	Number of lags	
	A	Number of LVs	
KPCA	k_{p}	Pairwise kernel function	
	σ or $\{c_0, d, \gamma\}$	Kernel parameters	$c_0 = 1$
	A	Number of PCs	
KPLS	k_{p}	Pairwise kernel function	
	σ or $\{c_0, d, \gamma\}$	Kernel parameters	$c_0 = 1$
	A	Number of LVs	
DKPCA	L	Number of lags	
	k_{p}	Pairwise kernel function	
	σ or $\{c_0, d, \gamma\}$	Kernel parameters	$c_0 = 1$
	A	Number of PCs	
DKPLS	L	Number of lags	
	k_{p}	Pairwise kernel function	
	σ or $\{c_0, d, \gamma\}$	Kernel parameters	$c_0 = 1$
	A	Number of LVs	
KDE-CVA	L	Extent of past horizon	
	H	Extent of future horizon	$H = L$
	A	Memory order	
	$\xi_{T_X^2}$	Scale factor for kernel width of T_X^2	
	ξ_{Q_X}	Scale factor for kernel width of Q_X	
	$\xi_{T_{X,\mathrm{r}}^2}$	Scale factor for kernel width of $T_{X,\mathrm{r}}^2$	$\xi_{T_{X,\mathrm{r}}^2} = \xi_{T_X^2}$
SVDD	k_{p}	Pairwise kernel function	
	σ or $\{c_0, d, \gamma\}$	Kernel parameters	$c_0 = 1$
	C	Radius-to-coverage parameter	

must be normal; the joint distributions of all couples of variables must be bivariate normal (Korkmaz et al., 2014; Oppong et al., 2016). For example, considering the case of an arbitrarily distributed variable nonlinearly correlated with a normal variable, its marginal distribution will be non-normal; furthermore, its joint distribution with any other variable will be non-normal.
Considering these facts, the criteria for nonlinearity and non-normality are expected to be in accordance most times. There are two special scenarios nonetheless:

- variables feature linear correlation only, but marginal distributions of some variables are non-normal, as when a variable is a linear combination of non-normal variables;
- variables are independent, but marginal distributions of some variables are non-normal, as in the case of the multivariate uniform distribution.

In both cases the dataset is non-normal and linear (in terms of correlation) and, according to Figure 7.2, a linear model is selected by SPAfPM (note that, in the second one of the cases mentioned above, linear and nonlinear models are expected to perform similarly due to independence of variables, the only consequence being that a large number of PCs/LVs/CVs is required). However, the performance in fault detection could still be disappointing due to scores being non-normally distributed. Therefore, KDE is suggested to estimate the control limits in order to adapt them to the actual distributions of the fault detection statistics.

7.3.3 A note on discrete variables

Discrete variables, such as categorical or binary variables, are common in industrial data as they can be used to mark process phases, process settings, or onsets of given process conditions (for example whether on on-off controller is acting or not). Furthermore, the limited measurement accuracy of some process sensors could imply that only few digits are recorded by the data acquisition system of the plant, causing continuous process variables to appear as varying on discrete levels. The presence of discrete variables requires special attention in the context of process monitoring.
As argued above, standard methods for fault detection are appropriate only if the available data feature linear correlation/are normally distributed, while kernel extensions can deal with nonlinear correlation/non-normal distributions. Discrete variables are clearly non-normally distributed (the normal distribution is defined for continuous variables), implying that nonlinear methods should be used if such variables are found in the dataset. However, even kernel-based methods are, in general, applicable to continuous variables only. Even fundamental concepts, such as the covariance, need special treatments in the presence of discrete or categorical variables (Niitsuma et al., 2005; Okada, 2000).
Some work has been done in the literature on the development of latent-variable models for categorical variables (Jollife et al., 2016). Considering PCA, the strategy is essentially to use nonlinear PCA (generally KPCA) combined with an optimization procedure to determine suitable numerical values to represent the levels of the discrete variables (Blasius et al., 2005;

Gower et al., 2005; Linting et al., 2007). In fact, kernel methods, especially the ones based on support vectors, proved to perform well in the presence of discrete variables (Pilario et al., 2020). However, the most common approach to deal with discrete variables is to simply encode them as binary values according to the dummy variable (or one-hot) encoding (Hastie et al., 2009), then directly include the resulting binary variables in the PCA (or KPCA) model (Tomba et al., 2013a, 2014). This approach could not be entirely appropriate.

A further concern with discrete variables regards the concept of dynamics, which is unclear, especially for qualitative variables (for example a variable assuming the values "red", "blue", and "green"). The appropriateness of autocorrelation to detect dynamics (see Section 7.4.3) is unclear as well in this case, due to its relationship with the concept of covariance.

Given the aforementioned points, the use of discrete variables in SPAfPM is highly discouraged. However, the inclusion of such variables is allowed anyway by the SPAfPM code, but only in the form of binary or integer variables. Qualitative variables need to be numerically encoded beforehand by the user. If discrete variables are included in the dataset, SPAfPM does not perform any dynamic assessment and the model pre-selection defaults to SVDD. Kernel-based SVDD is in fact known to perform well on data including discrete variables due to the ability of Gaussian kernels to deal with highly non-normal distributions, implicitly mapping them to normal distributions in the feature space (Choi et al., 2005; Cremers et al., 2003).

7.4 Preliminary data interrogation procedure

The criteria used to assess the relevant data characteristic introduced in the previous Section, that is, non-normality, nonlinearity, and dynamics, are introduced in this Section. The effectiveness of the criteria is demonstrated using rigorous Monte Carlo simulations. All the simulations are carried out in Python 3.9.12 (Python Software Foundation, 2022) and R 4.2.0 (R Foundation, 2022). The two environments are interfaced by means of rpy2 (rpy2, 2022).

7.4.1 Non-normality detection

Mecklin et al. (2005) carried out a Monte Carlo study to investigate the effectiveness of various multivariate normality tests and concluded that the Henze-Zirkler test (Henze et al., 1990) is to be preferred due to its better empirical performance and theoretical properties. They also found the Royston test (Royston, 1983) to perform very well, matching the performance of the Henze-Zirkler test. They finally pointed out that the Mardia skewness and kurtosis tests (Mardia, 1970) show good performance and are among the most widely used tests for multivariate normality.

The four testes mentioned above are considered to assess non-normality in SPAfPM. Preliminary analyses on the tests highlighted advantages and drawbacks of each one. Considerations regarding the theoretical foundations of the tests further backed up the empirical results. The most important outcomes of the preliminary assessments are as follows.

- The statistic used in the Henze-Zirkler test is based on the lognormal distribution and its variance shrinks to zero as the number of variables increases (unless balanced by a very large number of observations), which causes numerical errors to become larger and larger with the increasing number of variables, compromising the reliability of the test.
- The Royston test can be applied to datasets with up to 2000 observations due to its formulation relying on empirically determined factors.
- The Royston test and the two Mardia tests require the inversion of the sample covariance matrix of the data, therefore they cannot be applied if such a matrix is singular (for example, if there are more variables than observations).

Besides these preliminary considerations, a Monte Carlo study is carried out to properly evaluate the performance of the four tests. The factors considered in the Monte Carlo study are:

- the distribution used to generate the dataset, which can be: multivariate normal, multivariate t, multivariate lognormal, or multivariate uniform;
- the number of variables in the dataset: $V_X \in \{10, 30, 50, 100, 200\}$;
- the number of observations in the dataset: $N \in \{50, 200, 500, 1000, 3000\}$.

All possible combinations of the factors are explored. For each combination, 100 repetitions are performed. For each repetition, a dataset is generated from the selected distribution using randomly selected parameters of the sampling distribution (different at each repetition).

Four normality tests (the Henze-Zirkler test, the Royston test, the Mardia skewness test, and the Mardia kurtosis test) are performed on the generated dataset (note that, as the Royston test and the Mardia tests cannot be applied to datasets containing less observations than variables, the Henze-Zirkler test is also not applied in the relevant cases for consistency). The outcomes of the four tests are saved for each repetition of a given combination of factors and used to compute the non-normality detection rates of each one of the four tests. The non-normality detection rate of a given test is defined as the number of repetitions over which the dataset is deemed non-normal by the relevant tests divided by the total number of repetitions. The four non-normality detection rates are the responses of the Monte Carlo study. These detection rates should be as close as possible to the chosen significance level ($\alpha = 0.01$) for the multivariate normal distribution, and to its complementary to one ($\beta = 0.99$) for all the other distributions. The results of the four selected tests are further combined to yield two more responses for the Monte Carlo study, which are also reported in the discussion below and are defined as:

- results of the Mardia skewness and kurtosis tests are used to obtain the detection rate of the Mardia combined test (a dataset is deemed non-normal if either one of the two tests detects non-normality);
- results of the four tests are combined in the "overall" test described at the end of this Section.

A second Monte Carlo study is set up, modifying the dataset generation mechanism. The "sampling distribution" factor is replaced by the "fraction of nonlinear variables" factor. The

domain of such a factor is: $f_{\text{nl}} \in \{0, 0.05, 0.1, 0.2, 0.4, 0.8\}$. The dataset degeneration mechanism is illustrated by means of an example. Assume that $V_X = 25$, and that 30% of the variables are nonlinearly correlated ($f_{\text{nl}} = 0.3$) with the remaining 70% of variables, which can feature a varying degree of linear correlation among each other. The first step is to sample $V_X^{\text{lin}} = \lfloor 0.7 V_X \rfloor = 17$ variables from a multivariate normal distribution with randomly generated parameters. Then, $V_X^{\text{nl}} = V_X - V_X^{\text{lin}} = 8$ additional variables are generated by randomly picking V_X^{nl} out of the V_X^{lin} linear variables (with replacement, if $V_X^{\text{nl}} > V_X^{\text{lin}}$) and applying nonlinear transformations randomly selected from a library of 60 nonlinear transformations. White noise is added to each one of the V_X^{nl} nonlinear variables by sampling independent normal distributions with zero means and variances selected so that the signal-to-noise ratio of each of the transformed variables is $1{:}\,0.1$. Finally, the V_X^{lin} linear variables and the V_X^{nl} nonlinear variables are jointed to produce the dataset. Responses of the second Monte Carlo study are the same of the first Monte Carlo study.

Results of the first Monte Carlo study on detection of normality are shown in Figure 7.3. The Royston test performs the best overall, always yielding non-normality detection rates very close to the nominal significance level. The Henze-Zirkler test is nearly equivalent in terms of performance for most cases; however, its performance visibly deteriorates when the dataset includes more than 50 variables (non-normality is detected by default as the test statistic is stuck to its maximum value, which causes the p-value to be always 0). Such behavior is due to the aforementioned variance shrinkage of the lognormal distribution used to compute the test statistic. The Mardia skewness test also performs well, but the Mardia kurtosis test does not perform as well due to the inherent difficulty in properly characterizing the kurtosis of high-dimensional multivariate distributions, which requires a very large number of observations.

Considering results on other distributions (see Appendix A for details), we can draw the following conclusions.

- All tests yield nearly the same performance when applied to the multivariate lognormal distribution, which is highly non-normal.
- All tests yield nearly the same performance when applied to the multivariate t distribution, which is slightly non-normal and converges to a multivariate normal distribution for increasing degrees of freedom. The Henze-Zirkler test performs marginally better than others for small sample sizes, although it also exhibits more erratic results.
- The Royston test outperforms other tests on the multivariate uniform distribution. In particular, the Henze-Zirkler test yields very erratic results in this case, even for less than 50 variables.

All these observations are confirmed by the second Monte Carlo study, in which the number of nonlinear variables is manipulated rather than the whole distribution. The Royston test performs slightly better than the Henze-Zirkler test for mild deviations from normality ($f_{\text{nl}} = 0.05$ and

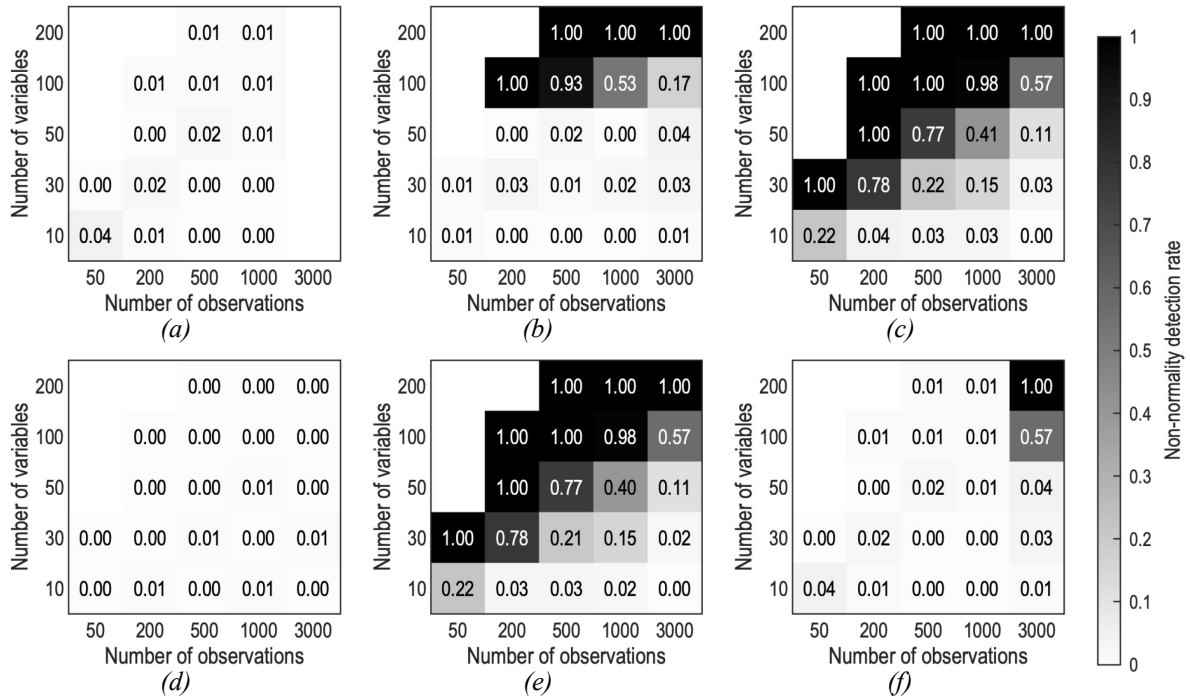


Figure 7.3. *Non-normality detection rates of multivariate normality tests on datasets generated from multivariate normal distributions: (a) Royston test, (b) Henze-Zirkler test, (c) Mardia combined test, (d) Mardia skewness test, (e) Mardia kurtosis test, and (f) combination of all the tests. Missing values mean that the relevant test is not applicable for a given combination of factors.*

$f_{\mathrm{nl}} = 0.1$), especially when the dataset does not include many observations. In this case, both the Mardia skewness and kurtosis tests yield erratic results, as in the case of $f_{\mathrm{nl}} = 0.2$ shown in Figure 7.4, which is also the case where the Royston test outperforms the Henze-Zirkler test most apparently, the latter exhibiting very erratic results. The performances of all tests converge for high fractions of nonlinear variables, where deviations from normality become apparent. Mild deviations from normality are hard to detect, as expected, especially on datasets with a small number of observations (see Appendix A for details).

Given these findings and bearing in mind the remarks made by Mecklin et al. (2005), the default criterion to test non-normality of the dataset in SPAfPM is selected as the Royston test, being the one that offers the best balance between performance and robustness over a wide range of cases. If the dataset includes more than 2000 observations, the Henze-Zirkler test is used when there are less than 51 variables in the dataset, and the combined Mardia test is used otherwise.

7.4.2 Nonlinearity detection

Unsatisfactory monitoring performance have been reported when PCA is applied to non-normal data (Zhu et al., 2016) due the mismatch between the data characteristics and the assumptions of the method. Non-normality of data can be induced by the presence of nonlinear correlation among variables, as argued in Section 7.3.2. Nonlinearity is therefore a relevant data characteristic assessed by SPAfPM.

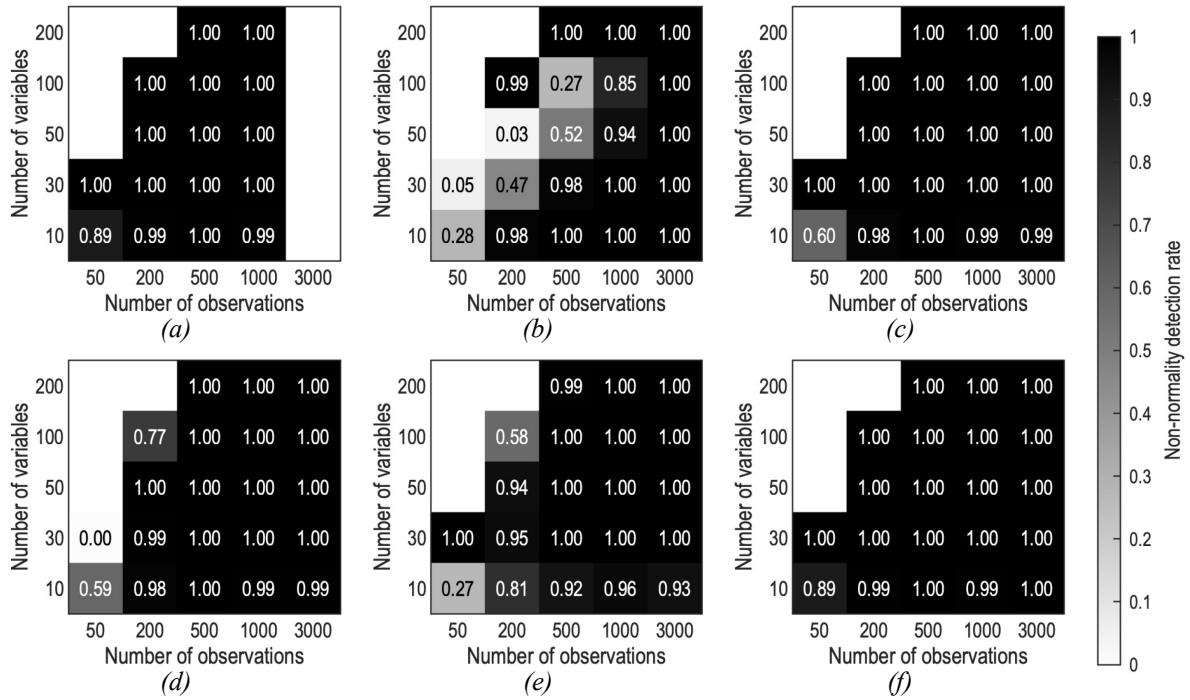


Figure 7.4. *Non-normality detection rates of multivariate normality tests on datasets in which 20% of the variables are nonlinear: (a) Royston test, (b) Henze-Zirkler test, (c) Mardia combined test, (d) Mardia skewness test, (e) Mardia kurtosis test, and (f) combination of all the tests. Missing values mean that the relevant test is not applicable for a given combination of factors.*

The proposed nonlinearity detection method is based on three tests performed simultaneously: linear correlation analysis (Montgomery et al., 2018), maximal correlation analysis (Rényi, 1959), and quadratic (correlation) test (Montgomery et al., 2018).

Given two vectors $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ gathering N observations of the random variables X and Y, respectively, the sample linear correlation coefficient (Montgomery et al., 2018) can be computed as:

$$r_{\mathbf{x},\mathbf{y}} = \frac{s_{\mathbf{x},\mathbf{y}}}{s_{\mathbf{x}} s_{\mathbf{y}}} = \frac{\frac{1}{N-1}\sum_{n=1}^{N}(x_n-\bar{x})(y_n-\bar{y})}{\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_n-\bar{x})^2}\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(y_n-\bar{y})^2}} \quad , \tag{7.37}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of X and Y, respectively, $s_{\mathbf{x}}$ and $s_{\mathbf{y}}$ are their sample standard deviations, and $s_{\mathbf{x},\mathbf{y}}$ is the sample covariance between X and Y. The linear correlation coefficient quantifies the degree of linear correlation between the two variables and varies between -1 and 1. Variables are uncorrelated if $r_{\mathbf{x},\mathbf{y}} \simeq 0$, while they are perfectly (anti-)correlated if $r_{\mathbf{x},\mathbf{y}} \simeq 1$ ($r_{\mathbf{x},\mathbf{y}} \simeq -1$).

The sample maximal correlation coefficient (Rényi, 1959) is defined as:

$$r^*_{\mathbf{x},\mathbf{y}} = \sup_{\rho,\psi}\left[r_{\rho(\mathbf{x}),\psi(\mathbf{y})}\right] \quad , \tag{7.38}$$

where ρ and ψ are functions from the set of all the measurable Borel functions with zero mean and are applied to $\mathbf{x}$ and $\mathbf{y}$ element-wise. Essentially, functions ρ and ψ are the nonlinear transformation that allow to maximize the linear correlation coefficient between the transformed variables $\rho(X)$ and $\psi(Y)$ (based on the available samples of the two random

variables), therefore operating an optimal linearization of the relationship between the original variables by explicit mapping. The sample maximal correlation coefficient can be computed by means of the alternating conditional expectation (ACE) algorithm (Breiman et al., 1985),which is also suitable to deal with discrete variables (for instance, categorical or binary) by default. The maximal correlation coefficient domain is $r^*_{\mathbf{x},\mathbf{y}} \in [0,1]$, where the transformed variables $\rho(X)$ and $\psi(Y)$ are uncorrelated if $r^*_{\mathbf{x},\mathbf{y}} \simeq 0$ and perfectly correlated if $r^*_{\mathbf{x},\mathbf{y}} \simeq 1$.
Comparing the absolute value of the linear correlation coefficient and the value of the maximal correlation coefficient provides an understanding of the nature of the relationship between X and Y:

- if $r_{\mathbf{x},\mathbf{y}} \simeq 0$ and $r^*_{\mathbf{x},\mathbf{y}} \simeq 0$, the variables are uncorrelated;
- if $r_{\mathbf{x},\mathbf{y}} \simeq 1$ and $r^*_{\mathbf{x},\mathbf{y}} \simeq 1$, the variables are linearly correlated (the functions ρ and ψ are both the identity function);
- if $r_{\mathbf{x},\mathbf{y}} \simeq 0$ and $r^*_{\mathbf{x},\mathbf{y}} \simeq 1$, the variables are nonlinearly correlated.

The quadratic test (Montgomery et al., 2018) searches for quadratic relationships between the random variables X and Y by first fitting two regression models, one linear and one quadratic, to the given samples, then comparing the performances of the two models by analysis of variances (ANOVA). The null hypothesis of the test is that the relationship between X and Y is linear (no significant difference between the linear and quadratic models), while the alternative hypothesis is that the relationship is quadratic. The null and alternative hypotheses are respectively formulated as:

$$H_0 : \mathbf{y} = b_{1_0}\mathbf{x} + b_{0_0} + \boldsymbol{\varepsilon}_0 \quad , \tag{7.39}$$

$$H_\mathrm{a} : \mathbf{y} = b_{2_\mathrm{a}}\mathbf{x}^2 + b_{1_\mathrm{a}}\mathbf{x} + b_{0_\mathrm{a}} + \boldsymbol{\varepsilon}_\mathrm{a} \quad , \tag{7.40}$$

where $\boldsymbol{\varepsilon}_0 \in \mathbb{R}^N$and $\boldsymbol{\varepsilon}_\mathrm{a} \in \mathbb{R}^N$ are samples of normal random variables. The F-test of the ANOVA can be applied for hypothesis testing, with the F-value computed from:

$$F_\mathrm{val} = \frac{\mathrm{MSE}_0 - \mathrm{MSE}_\mathrm{a}}{\mathrm{DOF}_0 - \mathrm{DOF}_\mathrm{a}} \left(\frac{\mathrm{MSE}_\mathrm{a}}{\mathrm{DOF}_\mathrm{a}}\right)^{-1} \quad , \tag{7.41}$$

where MSE_0 and MSE_a are the mean squared errors of the linear and quadratic models, respectively, and DOF_0 and DOF_a are the degrees of freedom of the two models. The test statistic is distributed as an F variable with $\mathrm{DOF}_0 - \mathrm{DOF}_\mathrm{a}$ numerator degrees of freedom and DOF_a denominator degrees of freedom, hence the p-value associated to the F-test can be computed as:

$$p_\mathrm{val} = 1 - F(\mathrm{DOF}_0 - \mathrm{DOF}_\mathrm{a}, \mathrm{DOF}_\mathrm{a})|_{F_\mathrm{val}} \quad , \tag{7.42}$$

where $F(\mathrm{DOF}_0 - \mathrm{DOF}_\mathrm{a}, \mathrm{DOF}_\mathrm{a})|_{F_\mathrm{val}}$ is the value of the inverse cumulative distribution function of the F variable evaluated at F_val. The p-value can then be compared to the significance level of the test adjusted by the Bonferroni correction (Hochberg, 1988), $\alpha_{\mathrm{QT}_\mathrm{adj}}$, to determine its significance. The quadratic correlation is deemed significant if $p_\mathrm{val} < \alpha_{\mathrm{QT}_\mathrm{adj}}$. The correction of the significance level is employed to control the false-positive rate when a large number of tests is performed simultaneously (Goeman et al., 2014; Nadon et al., 2002).

The nonlinearity assessment method used in our framework is based on the one originally proposed for SPA (Sun et al., 2021). The nonlinear correlation between a pair of variables is deemed significant if at least one of the following two test has a positive outcome.

1. The absolute value of the linear correlation coefficient is close to 0, while the maximal correlation coefficient is close to 1 (in other words, the coefficients differ significantly).
2. The p-value of the quadratic test is below the adjusted significance level.

Test 1 regards the significance of the difference between the maximal correlation coefficient and the absolute linear correlation coefficient. The test is based on two conditions:

- if $r^*_{\mathbf{x},\mathbf{y}} \leq \varepsilon_{\mathrm{MC}}$, the nonlinear correlation is deemed significant if $\left(r^*_{\mathbf{x},\mathbf{y}} - D\right) - \left|r_{\mathbf{x},\mathbf{y}}\right| > \varepsilon_1$, and insignificant otherwise (D is a correction factor discussed below);
- if $r^*_{\mathbf{x},\mathbf{y}} > \varepsilon_{\mathrm{MC}}$, the nonlinear correlation is deemed significant if $r^*_{\mathbf{x},\mathbf{y}} - \left|r_{\mathbf{x},\mathbf{y}}\right| > \varepsilon_2$, and insignificant otherwise.

Note that the two conditions are complementary (only one of them can be true at a time). Default values of thresholds are set as in SPA (Sun, 2020b): $\varepsilon_{\mathrm{MC}} = 0.92$, $\varepsilon_1 = 0.4$, and $\varepsilon_2 = 0.1$.

Test 2 regards the quadratic test. The nonlinear correlation is deemed significant if $p_{\mathrm{val}} < \alpha_{\mathrm{QT_{adj}}}$, and insignificant otherwise. The threshold for the test is:

$$\alpha_{\mathrm{QT_{adj}}} = \frac{\alpha_{\mathrm{QT}}}{C_{\mathrm{B_{QT}}}} \quad , \tag{7.43}$$

where α_{QT} is the nominal significance level of the test and $C_{\mathrm{B_{QT}}}$ is the Bonferroni correction factor. Such a correction is achieved by dividing the nominal significance level by the number of tests being performed simultaneously. If V_X variables are available in the dataset, then $V_X(V_X - 1)$ couples are to be tested (note that the quadratic test is not symmetric, unlike the correlation coefficient), therefore:

$$C_{\mathrm{B_{QT}}} = V_X(V_X - 1) \quad . \tag{7.44}$$

The default nominal significance level is set as in SPA (Sun, 2020b): $\alpha_{\mathrm{QT}} = 0.01$.

The first condition of test 1 involves a correction factor as well, D, that is subtracted to the value of the maximal correlation coefficient. The correction factor is introduced because the ACE algorithm used to estimate the maximal correlation coefficient is known to yield poor results when variables are nearly uncorrelated (Tibshirani, 1988). Preliminary tests carried out on uncorrelated variables highlighted that the maximal correlation coefficient tends to be "inflated" in this case, and that the magnitude of such an inflation depends on the number of observations in the dataset. The factor D is used to counteract this phenomenon[11].

The inflation of the maximal correlation coefficient is investigated by simulation. Specifically, $r^*_{\mathbf{x},\mathbf{y}}$ and $\left|r_{\mathbf{x},\mathbf{y}}\right|$ are computed for couples of uncorrelated normal variables with numbers of observations varying between 10 and 3000. For each number of observations, 100 couples are generated in order to compute medians and variabilities (as 99% percentile-based confidence limits) of the coefficients. Results are reported in Figure 7.5.

[11] Note that no correction is applied in the second condition of test 1, as the estimate is assumed to be reliable when $r^*_{\mathbf{x},\mathbf{y}}$ is high.

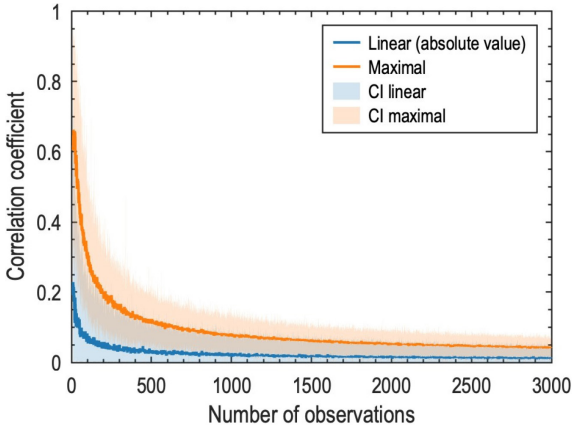


Figure 7.5. *Absolute linear correlation coefficient and maximal correlation coefficient for couples of uncorrelated normal variables. Solid lines are medians over 100 datasets; shaded areas delimit the percentile-based confidence intervals (CI) at 99% confidence level.*

The maximal correlation coefficient is observed to become inflated when the variables are independent, especially when the dataset features a small number of observations. The linear correlation coefficient suffers from a similar inflation, although to a much lesser extent. Furthermore, while the lower confidence limit of the linear correlation nearly touches 0 for all sample sizes, the limit of the maximal correlation coefficient is almost always higher than 0.

In absence of the correction factor D, the inflation of the maximal correlation coefficient would be propagated to the difference $r^*_{\mathbf{x},\mathbf{y}} - |r_{\mathbf{x},\mathbf{y}}|$, which could thus become greater that ε_1 by random chance. The correction factor is introduced to prevent this issue. Four options to set the value of the correction factor are enabled:

a. the correction factor is set to the median of the maximal correlation coefficient of two uncorrelated normal variables;
b. the correction factor is set to the 99% upper confidence limit of the maximal correlation coefficient of two independent variables;
c. the correction factor is set to the lower confidence interval at α significance level of the maximal correlation coefficient of the couple of variables being assessed, which is equivalent by replacing the value of the maximal correlation coefficient with its lower confidence limit;
d. no correction at all.

For approaches a and b, analytical functions for the deflation factor are fitted to the relevant data from the study shown in Figure 7.5. The functional form for fitting is $D = aN^b$, where a and b are parameters of the functional form. Results of the fittings reported in Figure 7.6.

On the other hand, approach c estimates the uncertainty on the maximal correlation coefficient of the actual variables considered by the test, which is done by bootstrap resampling (Efron, 1979; Efron et al., 1993). The bias-corrected and accelerated method (Efron, 1987) is used to counteract the overestimation of the correlation coefficient due to the resampling with replacement performed by the bootstrap. Note that approach c is statistically founded, as it relies

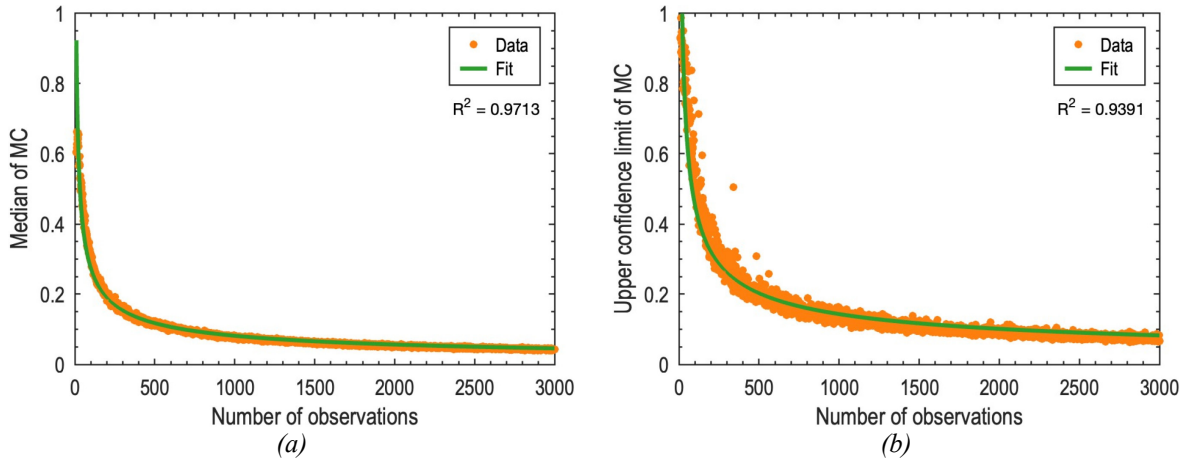


Figure 7.6. *Fittings used to determine the deflation factor in the assessment of significance of the nonlinear correlation: (a) median value of the maximal correlation coefficient over 100 samples (a = 3.1076 and b = −0.5269), and (b) 99% upper confidence limit of the maximal correlation coefficient over 100 samples (a = 4.5682 and b = −0.5011).*

on a theoretically sound uncertainty estimation procedure. Furthermore, this approach is general and not aimed at solving exclusively the issue with nearly uncorrelated variables. However, the computational cost of approach c is a very high due to the bootstrap resampling, especially when the number of variables in the dataset is large (the number of couples of variables to be assessed scales as V_X^2), hence approach a is the default in SPAfPM.

The nonlinearity assessment method described above (tests 1 and 2) and used in our framework is to be applied to all couples of variables in the dataset at hand. However, an aggregation rule is to be chosen, as the nonlinearity property must be assigned to the whole dataset rather than to specific couples of variables. Therefore, we consider the following three criteria for nonlinearity detection as candidates for the aggregation rule.

1. The "any" criterion: the dataset is deemed nonlinear if any couple of variables feature a significant nonlinear correlation, consistently with the SPA approach (Sun et al., 2021).
2. The "variables" criterion: the dataset is deemed nonlinear if the fraction of variables involved in a significant nonlinear relationship with another variable is greater than $\varepsilon_{\mathrm{nl}}$.
3. The "couples" criterion: the dataset is deemed nonlinear if the fraction of couples of variables featuring significant nonlinear correlation is greater than $\varepsilon_{\mathrm{nl}}$.

Note that the second and third criteria have a remarkable advantage over the first one. Since $V_X(V_X - 1)$ couples of variables are tested, the "any" criterion entails a non-negligible probability of incorrectly detecting nonlinearity in the dataset due to a single false positive, thus leading to the selection of an unnecessarily complex nonlinear model. The probability of this occurrence increases quadratically with V_X. Furthermore, linear models can manage mildly nonlinear datasets by including additional PCs/LVs/CVs; see discussions by Paluš et al. (1992), and Dong et al. (1996) for details. The default value of the fraction of nonlinear variables/couples to be used in both the second and third criteria is set as $\varepsilon_{\mathrm{nl}} = 0.1$, as this fraction starts to exceed mildly nonlinear behavior that can still be handled by linear models.

The three criteria are compared by means of two Monte Carlo simulations, identical in settings to the ones discussed in Section 7.4.1, with the same factors but responses being now the nonlinearity detection rates of the "any", "variables", and "couples" criteria. Note that the first study still considers the sampling distribution as one of the factors. Although it is not known *a priori* whether the considered distributions feature nonlinear correlation of variables or not, this study is done to test the hypothesis advanced in Section 7.3.2, namely that non-normality and nonlinearity are tightly interconnected properties of a dataset. For the same reason, the non-normality detection rate is also included among responses of the Monte Carlo studies on nonlinearity detection. The combination of non-normality detection tests according to the rationale outlined at the end of Section 7.4.1 is used to test non-normality. Settings of all criteria are kept to default values.

A general comment emerging from both Monte Carlo studies is that the number of observations is extremely important for the reliability of nonlinearity assessments. In fact, all criteria correctly deem datasets generated from normal distribution as normal in nearly all repetitions only for $N \geq 500$. Figure 7.7 shows that, as expected, the "any" criterion is the least robust one, while the "couples" criterion is the most robust one, being perfect in recognizing linear datasets even for $N \geq 200$. The "variables" criterion yields acceptable results for $N/V_X \geq 4$.

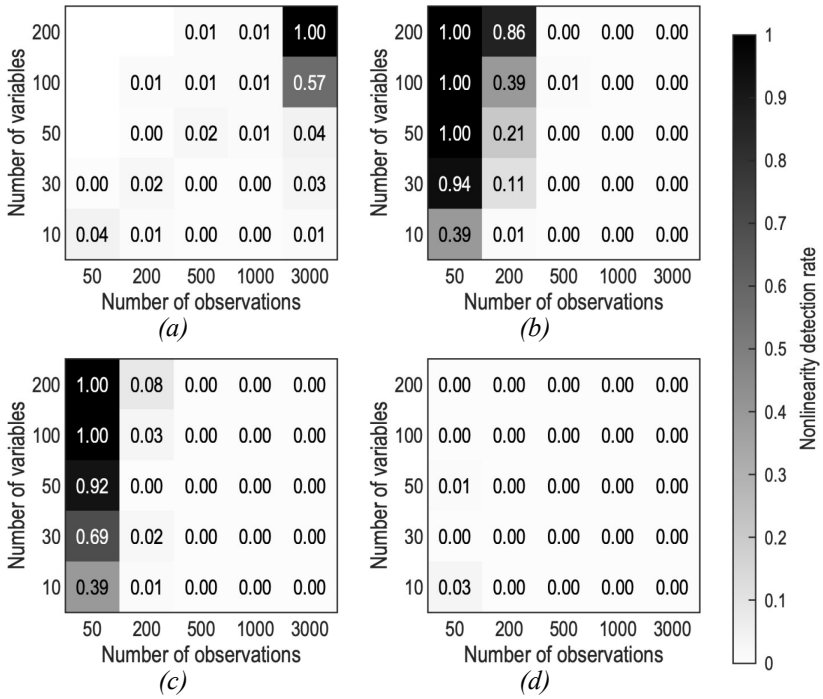


Figure 7.7. *Nonlinearity detection rates of the proposed criteria on datasets generated from multivariate normal distributions: (a) combination of non-normality tests, (b) "any" criterion, (c) "variables" criterion, and (d) "couples" criterion.*

Considering datasets generated from other distributions (see Appendix A for details), all criteria are nearly perfect in detecting nonlinearity of the lognormal distribution, with the "couples" criterion sporadically exhibiting erratic behavior. Detection of the multivariate t distribution is

harder, due to the mild deviation from normality. In this case, the "couples" criterion is the worst performing one, while the "any" criterion performs best. However, the performance of the "any" criterion on the multivariate t distribution could just be due to its lack of robustness. Results on the uniform distribution, shown in Figure 7.8, are particularly interesting. As expected, samples are correctly deemed non-normal and linear. This occurs due to the multivariate uniform distribution featuring no correlation at all. Such results show that the default thresholds for nonlinearity assessment regarding the maximal correlation coefficient, together with the default deflation approach, are appropriate to not misclassify independent variables as nonlinearly correlated. The results also confirm that at least 500 observations are needed for the reliability of the "any" criterion, whereas the "variables" and "couples" criteria allow to lower that threshold to $N \geq 200$, though a larger number of observations is still recommended to obtain high reliability of nonlinearity detection.

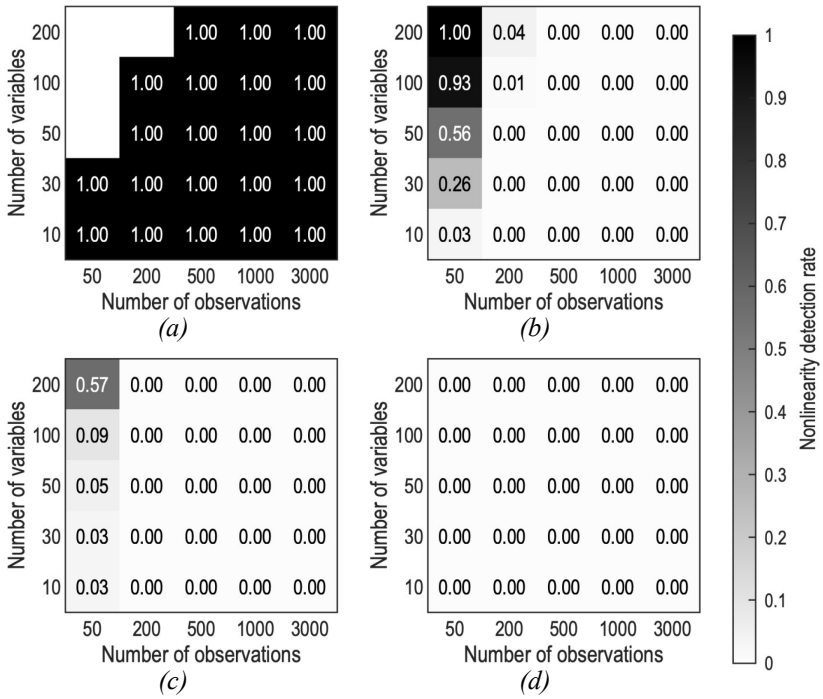


Figure 7.8. *Nonlinearity detection rates of the proposed criteria on datasets generated from multivariate uniform distributions: (a) combination of non-normality tests, (b) "any" criterion, (c) "variables" criterion, and (d) "couples" criterion.*

Overall, the results of the first Monte Carlo study prove the validity of the discussion outlined in Section 7.3.2: the non-normal distribution of data and the presence of nonlinear correlation among variables are equivalent, saved special cases as the multivariate uniform distribution.
Moving to the Monte Carlo study generating datasets given the fraction of nonlinear variables, consider the case $f_{\mathrm{nl}} = 0.05$ first. In this case, no nonlinear variable is included if $V_X = 10$, while only one variable is included if $V_X = 30$. This last occurrence yields the minimum value of the fraction of nonlinear variables, in this case $2/V_X = 0.06667$, achieved if one single couple features nonlinear correlation. Figure 7.9 highlights the importance of the ratio of the

number of observations to the number of variables, that is especially apparent from the results of the "variables" criterion. Finally, the "couples" criterion is the only one consistently recognizing the dataset as linear according to the set threshold $\varepsilon_{\mathrm{nl}}$.

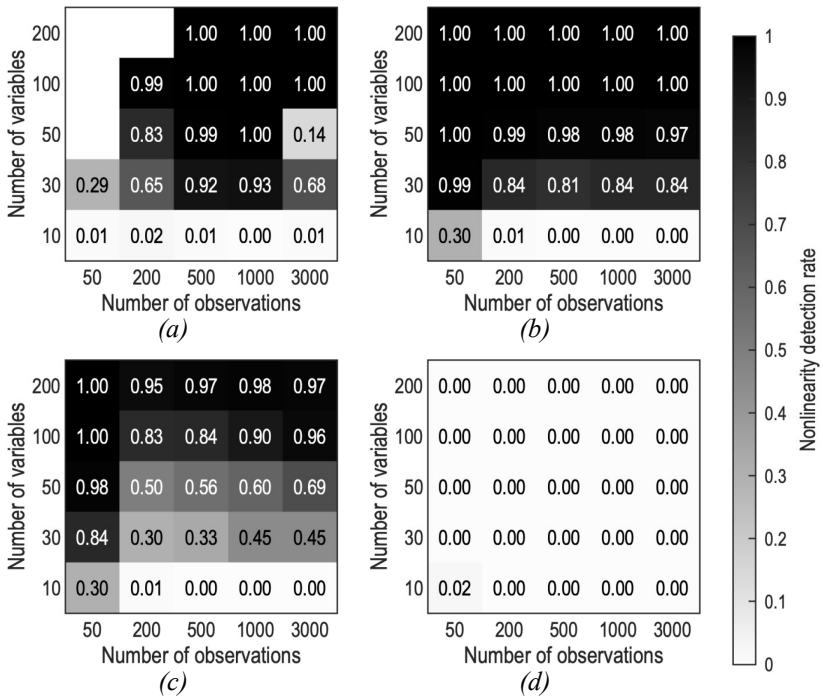


Figure 7.9. *Nonlinearity detection rates of the proposed criteria on datasets in which 5% of the variables are nonlinear: (a) combination of non-normality tests, (b) "any" criterion, (c) "variables" criterion, and (d) "couples" criterion.*

Cases with higher values of f_{nl} allow us to draw conclusions similar to the those already known concerning the robustness of methods. Besides the case $f_{\mathrm{nl}} = 0.1$, where the "any" criterion appears to be a little too strict with respect to the "variables" criterion (the former has detection rates always very close to 1 even for low f_{nl}), these two criteria show similar results (see Appendix A for details). On the other hand, the "couples" criterion consistently misses the nonlinearity of the dataset, achieving acceptable performance only if $N/V_X \geq 100$, which is unreasonable. This lack of performance could be due to the fact that the number of couples required to overtake the threshold for this criterion varies as V_X^2, thus increasing sharply with the number of variables. This makes the criterion robust to the rejection of the nonlinearity hypothesis, but overly conservative to its acceptance, therefore being prone to high false-negative rates. The case with $f_{\mathrm{nl}} = 0.4$ is shown in Figure 7.10 as an example of this behavior. Considering all the outcomes of the Monte Carlo studies, the "variables" criterion is chosen as the default criterion to assess nonlinearity of a dataset. The motivation is that this method shows the best tradeoff between detection rate on nonlinear datasets and rejection rate on linear datasets, being sufficiently robust and sensitive for $N \geq 200$ and $N/V_X \geq 4$. Furthermore, this method offers a nice insight on the "intensity" of the nonlinearity of the dataset, which can be quantified by the fraction of variables involved in nonlinear relationships. The most

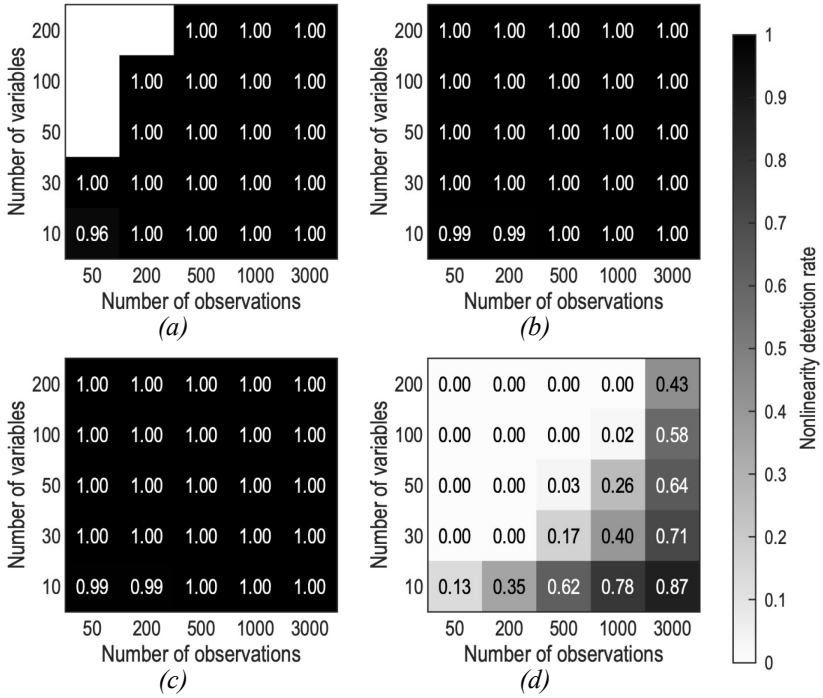


Figure 7.10. *Nonlinearity detection rates of the proposed criteria on datasets in which 40% of the variables are nonlinear: (a) combination of non-normality tests, (b) "any" criterion, (c) "variables" criterion, and (d) "couples" criterion.*

prominent drawback of the selected methods is that its resolution (minimum value that the fraction of nonlinear variables can assume) degrades as the number of variables decreases.

7.4.3 Dynamics detection

Dynamics is a relevant data characteristic in SPAfPM as basic methods, such as PCA and PLS, could not be able to detect faults affecting process dynamics, as they assume that data do not feature any autocorrelation (Ku et al., 1995). Dealing with multivariable random processes (time-dependent random vectors), three functions are helpful to characterize dynamics (Box et al., 2016): the ACF characterizes the general dynamic behavior of a time series; the partial autocorrelation function (PACF) characterizes the dynamics of a time series in terms of optimal autoregressive models, thereby "removing" the effect of the ACF; the cross-correlation function (CCF) characterizes the interdependence between the dynamics of two time series. In their sample versions, these functions exploit the concept of lagged measurements introduced in Section 7.2.3 and yield one coefficient for each lag. The significance of coefficients can be tested with the normal deviate approach, where insignificant coefficients are assumed to follow a normal distribution with zero mean and variance estimated by either the large sample approximation (Box et al., 2016; Quenouille, 1949) or the lag-corrected estimator (Chatfiled et al., 2019). For the significance of ACF coefficients, Bartlett's formula (Bartlett, 1946) offers an additional variance estimator, and the Ljung-Box statistic (Ljung et al., 1978) is an alternative approach to significance assessment, which furthermore allows to adjust the nominal

significance level using the Bonferroni correction (Hochberg, 1988), if multiple coefficients are tested simultaneously. In general, a variable features no significant dynamics if no coefficient is deemed as significant in either the ACF or PACF, while two time series are uncorrelated if no significant coefficient is found in the CCF.

Preliminary tests (not shown here for brevity) were performed to investigate the behavior of the ACF and PACF for dynamics detection in single static and dynamic variables. All possible combinations of significance assessment approach, variance estimator, and significance level adjustment were considered for different numbers of lags tested simultaneously. The best results are obtained using the ACF alone, testing coefficients corresponding to $L = \min\{20, \lfloor N/2 \rfloor - 1\}$ lags with the Ljung-Box approach (which is independent on the variance estimator) and Bonferroni correction. The PACF should not be used, as it did not achieve a satisfying balance between robustness and sensitivity in the tests carried out. The CCF should not be used either, as its robustness was very low: for a dataset with V_X variables, $V_X(V_X - 1)$ couples of variables are assessed by CCF, and the risk of mistakenly deeming a non-negligible number of insignificant CCF coefficients as significant increases sharply.

Given a time series $\mathbf{x} \in \mathbb{R}^N$ of a random process X, the (sample) ACF coefficient at lag l is defined as (Box et al., 2016):

$$r_{\mathbf{x}}(l) = \frac{c_{\mathbf{x}}(l)}{c_{\mathbf{x}}(0)} \quad , \tag{7.45}$$

where $c_{\mathbf{x}}(l)$ is the sample autocovariance function of the time series at lag l, defined as:

$$c_{\mathbf{x}}(l) = \frac{1}{N}\sum_{n=1}^{N-l}(x_n - \bar{x})(x_{n+l} - \bar{x}) \quad , \tag{7.46}$$

where $\bar{x}$ is the sample mean of the process[12]. The significance of autocorrelation coefficients can be determined using the Ljung-Box statistic (Ljung et al., 1978):

$$\tilde{Q}(l) = N(N+2)\sum_{j=1}^{l}\frac{\left(r_{\mathbf{x}}(j)\right)^2}{N-j} \quad . \tag{7.47}$$

The $\tilde{Q}(l)$ statistic is approximately distributed as a χ^2 variable with l degrees of freedom. The p-value of the statistic can be compared to the adjusted significance level of the test:

$$\alpha_{\mathrm{ACF_{adj}}} = \frac{\alpha_{\mathrm{ACF}}}{C_{\mathrm{B_{ACF}}}} \quad , \tag{7.48}$$

where the Bonferroni correction factor is $C_{\mathrm{B_{ACF}}} = l$, the number of coefficients being tested simultaneously. By default, we set $\alpha_{\mathrm{ACF}} = 0.01$ as in SPA (Sun, 2020b). The sample $\mathbf{x}$ is deemed to feature significant dynamics if at least one coefficient $r_{\mathbf{x}}(l)$, $l \in \{1, \dots, L\}$, is found significant. The dynamics test is not performed on discrete variables (see Section 7.3.3).

Similarly to the nonlinearity detection, dynamics is tested on all variables, but is to be attributed to the whole dataset. The following two criteria are proposed to detect dynamics in a dataset of N observations and V_X variables.

1. The "any" criterion: the dataset is deemed dynamic if any of the variables feature a significant dynamic behavior.

[12] Aside: $c_{\mathbf{x}}(0)$ is a biased version of the sample variance ($s_{\mathbf{x}}^2$) of the process.

2. The "variables" criterion: the dataset is deemed dynamic if the fraction of variables featuring significant dynamics is greater than ε_{dyn}.

We expect the "variables" criterion to offer advantages over the "any" criterion in terms of robustness (see Section 7.4.2). However, the risk to mistakenly detecting dynamics in the dataset due to a single false positive now scales linearly with V_X, although such an occurrence would nonetheless lead to the selection of a dynamic model when a static model would be more appropriate. Static models are also able to capture mild dynamics by including additional PCs/LVs, as argued by Vanhatalo et al. (2016) and proved by the study in Chapter 4. The default value of the fraction of dynamic variables for the relevant criterion is set as $\varepsilon_{\text{dyn}} = 0.1$.

In contrast to constructing regression models, where dynamics is tested on residuals of a static regression model (Sun et al., 2021), in the proposed framework dynamics is assessed directly on the variables in the dataset. As argued by Ku et al. (1995), applying a static model to dynamic data can extract only static components, with the dynamics being left in the residual space. In such a case, the Q_X statistic is expected to carry the dynamics of the residuals, hence featuring significant autocorrelation. This point can be leveraged to propose two additional criteria for dynamics detection. First, a static model of choice is selected according to the outcomes of the nonlinearity detection criterion and the presence of dependent variables. Two versions of the model are built: model A using the parameters corresponding to the minimum error in cross-validation; model B using the one-standard-error-rule (see Section 7.5.3 for details). Based on this, two additional criteria for dynamics detection can be defined as follows.

3. The "model_min" criterion: the dataset is deemed dynamic if the Q_X statistic from model A features significant dynamics;
4. The "model_oster" criterion: the same rationale of the previous criterion is adopted, but Q_X coming from model B.

The four proposed criteria are evaluated in a Monte Carlo study. The factors of the study are:

- the fraction of dynamic variables in the dataset: $f_{\text{dyn}} \in \{0, 0.05, 0.1, 0.2, 0.4, 0.8\}$;
- the number of variables in the dataset: $V_X \in \{10, 30, 50, 100, 200\}$;
- the number of observations in the dataset: $N \in \{50, 200, 500, 1000, 3000\}$.

All combinations of factors are tested, and 100 repetitions are performed for each combination, generating a random dataset at each repetition. The dataset is generated in a similar way as described in Section 7.4.1. Assume, for example, that $V_X = 25$ and that 30% of the variables are dynamic ($f_{\text{dyn}} = 0.3$), while the remaining 70% are static variables. The first step is to sample $V_X^{\text{sta}} = \lfloor 0.7 V_X \rfloor = 17$ variables from a multivariate normal distribution with randomly generated parameters. Then $V_X^{\text{dyn}} = V_X - V_X^{\text{sta}} = 8$ dynamic variables are to be generated. Matrices of a random state-space model are generated using an algorithm inspired by the `drss` method provided by the Systems Identification Toolbox (The Mathworks, 2022b) of MATLAB R2022a (The Mathworks, 2022a). The generation of the matrices of the state-space model is tuned in a way that guarantees stability of the system and that the feed-through matrix is null

(no direct effect of current inputs on current outputs). The state-space model has V_X^{sta} inputs and V_X^{dyn} outputs, with the number of states randomly selected as an integer between 1 and 10 (inclusive). The observations of the V_X^{sta} variables are used as inputs to run the state-space model, while the corresponding outputs are collected as the V_X^{dyn} dynamic variables. In order to simulate stationary processes (assumption of all the aforementioned significance assessment approaches), the state is randomly initialized and 200 more observations are sampled from the same distribution used to generate the V_X^{sta} static variables. Such observations are used to "burn-in" the state-space model with the randomly initialized state to obtain a stationary initial state, which is then used to generate the actual N observations of dynamic variables. Outputs of the 200 burn-in observations are discarded. White noise is added to each of the V_X^{dyn} dynamic variables by sampling independent normal distributions with zero means and variances selected so that the signal-to-noise ratio of each of the generated variables is $1:0.1$. Finally, the V_X^{sta} static variables and the V_X^{dyn} dynamic variables are jointed to produce the dataset.

Concerning the "any" and "variables" criteria, the false-positive rate of the former is higher in the cases $f_{\mathrm{dyn}} = 0$ (see Figure 7.11) and $f_{\mathrm{dyn}} = 0.05$, while the latter consistently deems the dataset as static with false-positive rate very close to the nominal significance level set for the ACF. Both the model-based criteria show good performance as well.

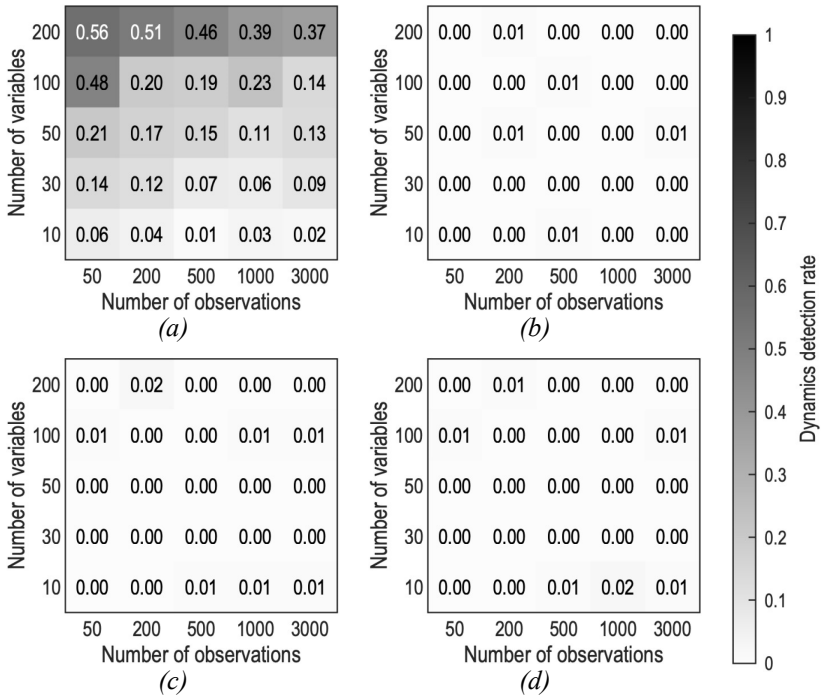


Figure 7.11. *Dynamics detection rates of the proposed criteria on samples in which 0% of the variables are dynamic: (a) "any" criterion, (b) "variables" criterion, (c) "model_min" criterion, and (d) "model_oster" criterion.*

The case $f_{\mathrm{dyn}} = 0.1$ shows a divergence in performances of the "any" and "variables" criteria, as can be seen in Figure 7.12: while the "any" criterion mostly deems datasets as dynamic, the "variables" criterion prefers static models, showing erratic dynamics detection rates. This

behavior is expected as $f_{\mathrm{dyn}} = 0.1$ is the threshold set for $\varepsilon_{\mathrm{dyn}}$. The two model-based criteria show again similar performance to the "variables" criterion yet yielding slightly more erratic results (no clear effect of the numbers of observations and of variables).

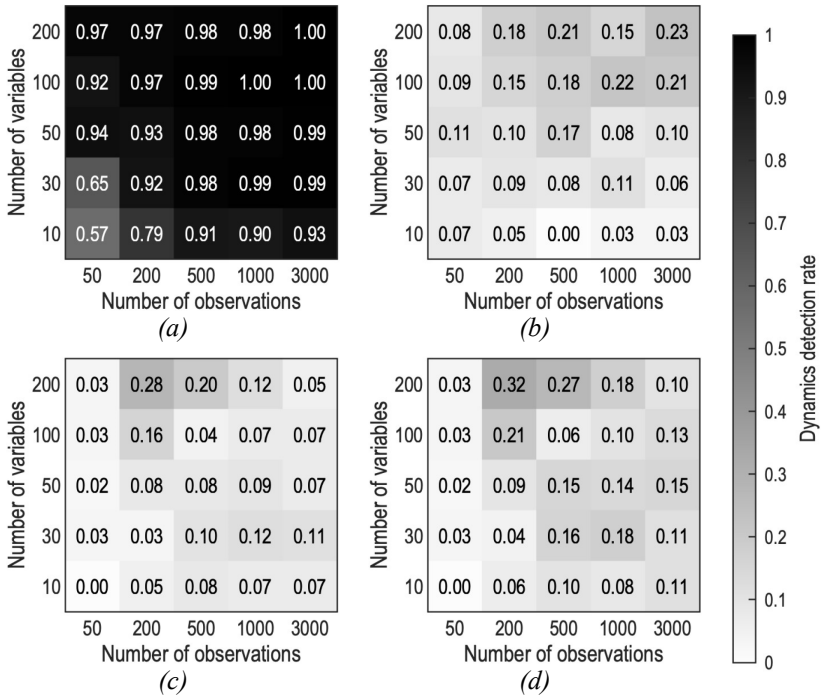


Figure 7.12. *Dynamics detection rates of the proposed criteria on samples in which 10% of the variables are dynamic: (a) "any" criterion, (b) "variables" criterion, (c) "model_min" criterion, and (d) "model_oster" criterion.*

The "any" and "variables" criteria show similar performance in the remaining cases (see Appendix A for details), with the latter being slightly more prone to deem the dataset as static than the former criterion for mild dynamics. On the other hand, both the model-based criteria show very high false-negative rates and quite erratic results. For example, Figure 7.13 reports the case $f_{\mathrm{dyn}} = 0.4$. These cases also show that long time series are required to properly characterize the dynamics in the data. A general indication is $N \geq 500$.

Based on these outcomes, the "variables" criterion is selected as the default dynamics assessment method of SPAfPM. As for the analogous criterion for nonlinearity, this criterion achieves the best tradeoff between robustness and sensitivity, while also offering a nice interpretation. The criterion is subject to the same drawback nonetheless, namely, poorer resolution as the number of variables decreases. A number of observations $N \geq 500$ is recommended for reliability of the dynamics detection test.

7.5 Model selection and discrimination procedure

The data analytics triangle in Figure 7.2 elucidates the selection process performed by SPAfPM to determine the most suitable model, or subset of models, based on the characteristics of the

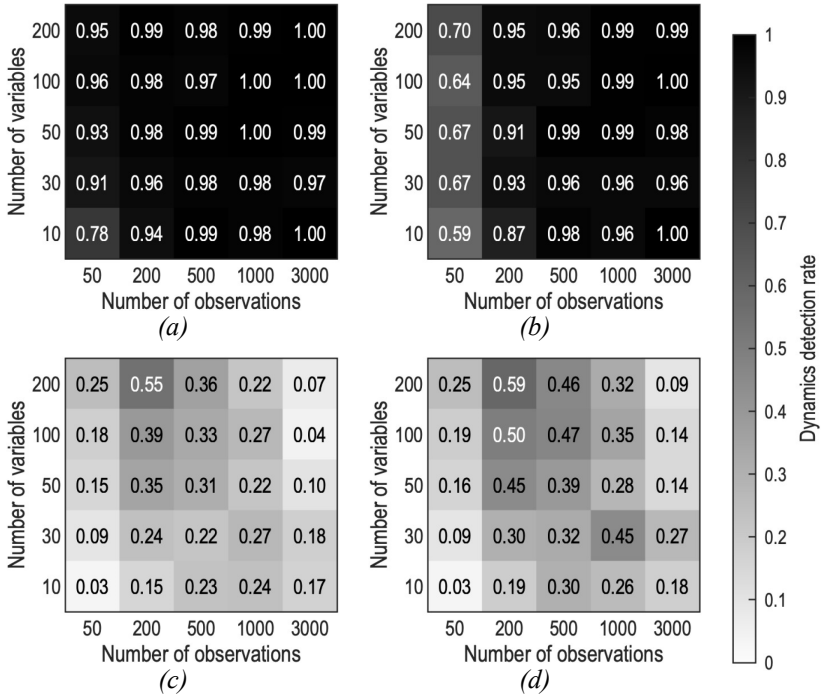


Figure 7.13. *Dynamics detection rates of the proposed criteria on samples in which 40% of the variables are dynamic: (a) "any" criterion, (b) "variables" criterion, (c) "model_min" criterion, and (d) "model_oster" criterion.*

data at hand. In case a subset of models is recommended, there needs to be a procedure to determine which one is the best (model discrimination). Additionally, the optimal hyperparameters for each one of the candidate models need to be tuned (model selection) to provide a fair comparison between them. A commonly used method to tackle both these problems is cross-validation (Allen, 1974; Stone, 1974).

7.5.1 Model selection in fault detection

Cross-validation is well established for model selection and discrimination. Considering, for instance, the case of regression, prediction performance of various models on a validation dataset (meaning data not used for model calibration) can be evaluated using the MSE as performance index (Sun et al., 2021). Similarly, the accuracy of a model can be evaluated on validation datasets as a measure of performance in a supervised classification problem (Mohr et al., 2019). However, in the case of fault detection, it is not trivial to define a good figure of merit to quantify the performance of a model (Camacho et al., 2014).

This problem can be tackled bearing in mind that the aim of model selection is to optimize the generalization performance of the model, therefore the performance index that is used should be consistent with the modeling objective (Camacho et al., 2014). Typically, the performance of a fault detection model is evaluated on how often is incorrectly qualifies NOC observations as faults (Type I error rate) and how often it misses faulty observations (Type II error rate). If the Type II error rate is to be used as a figure of merit, data from faulty operating conditions

must be available at the time of model calibration. While it is not difficult to produce such data using simulators, it is uncommon that comprehensive datasets including all possible faults are available in real, industrial applications (even though this might be the case for some specific processes). Therefore, SPAfPM relies on the restrictive assumption that only NOC data are available for model calibration and selection.

The Type I error rate can be used as a model evaluation metric as well. In fact, this is a "good practice" frequently mentioned in the fault detection literature: the validation Type I error rate (defined as that fraction of normal observations detected as faulty on a validation NOC dataset) should be as close as possible to the nominal significance level used to estimate control limits, α. This point is explicitly suggested by several studies (Camacho et al., 2016, 2006b; Ramaker et al., 2006; Yoon et al., 2004). To mention some examples, Ramaker et al. (2006) state that «it is useful to check whether the fraction of out-of-control signals for a given data set is close to α in case the control charts are set at this significance level. […] The performance of a chart in terms of Type I error is good if α observed is close to α», while Yoon et al. (2004) suggest that «By calculating the false alarm rate during normal operating conditions for the testing set and comparing it against the level of significance upon which the threshold is based, one can measure the robustness of a fault detection method». This condition is also regarded as essential when the performances of multiple fault detection models are to be compared (Camacho et al., 2009; Rato et al., 2013; Reis et al., 2021a). However, note that while all of the aforementioned studies suggest to match the Type I error to the nominal significance level by manual adjustment of control limits, none of the studies offers any guideline on how to select the hyperparameters of the relevant model, such as the number of PCs, consistently with the modeling objective in the sense of Camacho et al. (2014).

These points suggest that Type I error rate can be used as an evaluation metric for model selection consistently with the objective of fault detection. Ideally, the Type I error rate should be as close to α as possible. This strategy is actually known in the literature on model-aided adulteration detection (in which the objective is basically the same as of fault detection in industrial systems), where it is referred to as rigorous model selection approach (Rodionova et al., 2016). We ultimately want to choose the model and hyperparameters that yields a Type I error rate as close to α as possible. Performing model selection on the basis of the absolute deviation of the validation Type I error rate from α is consistent with the monitoring objective, as suggested by Camacho et al. (2014), and automates the fulfilment of the "criterion for good monitoring performance" suggested by Camacho et al. (2016), Ramaker et al. (2006), and Yoon et al. (2004). In this way, an empirical, possibly inconsistent model selection followed by an empirical adjustment of control limits is automated in a single, consistent operation.

The mathematical formulation of the model performance measure for fault detection is illustrated taking PCA as an example. In PCA, a Type I error occurs if either one of the T_X^2 or the Q_X statistics crosses the relevant control limit. Therefore, minimizing the deviation of the

validation Type I error rate from α is equivalent to minimizing the function:

$$J_\alpha^{\mathrm{PCA}} = \left| \frac{\sum_{n=1}^{N_{\mathrm{val}}} g_\alpha^{\mathrm{PCA}}(\mathbf{x}_n)}{N_{\mathrm{val}}} - \alpha \right| \quad , \tag{7.49}$$

where N_{val} is the number of observations in a validation dataset, $\mathbf{x}_n$ is the n-th observation in the same dataset, and g_α^{PCA} is the fault indicator function for PCA, defined as:

$$g_\alpha^{\mathrm{PCA}}(\mathbf{x}_n) = \begin{cases} 0 & \text{if } T_X^2(\mathbf{x}_n) \leq T_{X,\mathrm{lim}}^2|_\alpha \textbf{ and } Q_X(\mathbf{x}_n) \leq Q_{X,\mathrm{lim}}|_\alpha \\ 1 & \text{if } T_X^2(\mathbf{x}_n) > T_{X,\mathrm{lim}}^2|_\alpha \textbf{ or } Q_X(\mathbf{x}_n) > Q_{X,\mathrm{lim}}|_\alpha \end{cases} \quad . \tag{7.50}$$

Most of the models included in SPAfPM share this indicator function, being based on the same fault detection statistics. The fault indicator functions of SVDD and CVA-based approaches differ slightly due to the detection statistics of such models being different. The rationale of the fault indicator function is the same nonetheless: a fault is detected when any of the fault detection statistics crosses the relevant control limit.

However, it is worth highlighting a drawback of the performance index formulated as in (7.49). The first term within the absolute value sign defines the validation Type I error rate as the ratio of two integers. The resolution of the Type I error rate (minimum non-zero value it can achieve) is controlled by the number of observations in the validation dataset, being equal to $1/N_{\mathrm{val}}$. As the significance level is usually a small number, typically $\alpha = 0.05$ or $\alpha = 0.01$, the number of observations in the validation dataset must be large enough to guarantee a good resolution of the Type I error rat, thus being comparable to α. As a rule of thumb, $N_{\mathrm{val}} \geq 100$ should be guaranteed, which yields a resolution of the Type I error rate equal to at least 0.01. Larger values of N_{val} are preferable nonetheless.

7.5.2 Tailoring model selection to the characteristics of the data

Being cross-validation a model selection approach based on resampling of the observations in the dataset, one must be careful to not break the potential correlation structure among observations. For static models, a repeated k-fold cross-validation (Burman, 1989) procedure can be used. In this case, the data are randomly split into k sets (called "folds") containing roughly N/k observations each. Subsequently, $k-1$ sets are used to calibrate the model, while the remaining set is used as validation dataset. Each one of the k sets is used as validation dataset once as to use each one of the observations in the original dataset in guise of validation observations, hence k models are calibrated and applied to the k sets to obtain k "independent" values of the performance index defined in (7.49). Furthermore, repeated k-fold implies that the splitting procedure is repeated several times for different k-fold splits of the original dataset. If r repeats are performed, rk values of the performance index are available upon completion of the procedure, which can be used to estimate both its average value and variability for decision making. In general, $k = 5$ folds and $r = 10$ repeats are regarded to be appropriate to ensure the statistical reliability of the procedure (Breiman et al., 1992; Kim, 2009; Kohavi, 1995), hence these are the default values used in SPAfPM.

Repeated k-fold cross-validation cannot be applied to dynamic data because the random splitting results in the loss of the correlation structure among observations (Bergmeir et al., 2012). Therefore, the so-called growing window cross-validation (Makridakis, 1990) is employed when dynamic models are selected by SPAfPM. Data are first split into k blocks of contiguous observations, with no alteration in their order. At the first iterations, the first block is used to build a model, and the second block is used as the validation dataset. For the second iteration, the first two blocks are used to calibrate a model, and the third block is used for validation. The procedure is repeated until the $k-1$ blocks after the first one have been used in validation once, thus $k-1$ values of the performance index defined in (7.49) are obtained and used for model selection. By default, SPAfPM splits the data in $k=5$ blocks, coherently with the number of folds for static cross-validation. Note that cross-validation for dynamic data inherently results in a small number of values of the performance index ($k-1$, as opposed to the rk available for static data), which are furthermore obtained from models calibrated on a different number of observations at each iteration (due to the growing window scheme). This usually implies a higher variability of the values of the performance index.

It is crucial to note that the number of observations in the validation dataset, N_{val} in (7.49), is determined by k in both the cross-validation schemes described in this section, being roughly $N_{\mathrm{val}} \simeq N/k$. Therefore, increasing k may severely degrade the resolution of the Type I error rate, compromising the reliability of the performance index. In order to respect the rule of thumb outlined at the end of the previous Section, $N_{\mathrm{val}} \geq 100$, the calibration dataset should include a number of observations $N \geq 500$ if the default $k=5$ is used.

7.5.3 Hyperparameter tuning and model discrimination

In SPAfPM, cross-validation is used both for hyperparameter tuning (model selection) and to select the best model among the candidates proposed by the preliminary data interrogation procedure (model discrimination), if more than one model suits the data characteristics (see Figure 7.2). This marks an important difference between the selection mechanism of SPAfPM and the rationale of AutoML packages (Hutter et al., 2019): the additional screening step, based on the characteristics of the data at hand, ensures that only appropriate models are compared by cross-validation, therefore effectively limiting the chances of overfitting.

Hyperparameters of the candidate models are first optimized using the most appropriate cross-validation scheme, as discusses in the previous Section. The one-standard-error rule (Filzmoser et al., 2009; Hastie et al., 2009) is applied in order to increase model robustness: instead of just choosing the set of hyperparameters that yields the minimum value for the function in (7.49), the set of hyperparameters yielding the most parsimonious model with performance metric still within one standard error from the minimum value is chosen. Usually, this approach selects robust models that are less prone to overfitting (Sun et al., 2021). Finally, the model with the best cross-validation performance is designated as the best candidate for the given dataset.

7.5.4 Rigorous and compliant model selection

The procedure outlined in Section 7.5.1 sets up the model selection mechanism of SPAfPM according to a rigorous approach, as defined by Rodionova et al. (2016). Only NOC data are used in rigorous model selection, and the objective is to select the model with the Type I error rate closest to α. On the other hand, a compliant approach to model selection relies on a validation dataset including faulty conditions. In this case, the model is calibrated on NOC data and applied to the faulty dataset: the model with the Type II error rate closest to $\beta = 1 - \alpha$ is selected as the best one.

While adopting a compliant approach may guarantee good fault detection performance, it also entails substantial drawbacks. A comprehensive database of several (possibly all) faults must be available. While this may be the case for specific units in some processes, it is not a common occurrence in general. Even in cases where such a database exists, only one fault must be used as validation dataset, and different models could be selected when different faults are used[13], a point already reported in the literature. For example, in the words of Paredes et al. (2023): «Prior consideration of fault information […] brings substantial problems when characterizing the method's detection properties, as they become dependent upon the faults that were used during training». Furthermore, the Type I error rate of the model is disregarded and could thus be arbitrarily far from α, compromising model robustness; on the other hand, adjusting the model also considering the deviation of the Type I error rate from α could conflict with the objective of compliant model selection. Subtler drawbacks exist: even though the model would be optimized on a set of known faults, no guarantee is given about performance on unknown faults; the compliant model selection objective is discrimination between NOC and faulty conditions, which is inconsistent with the fault detection principle of describing NOC data. These drawbacks support the idea that relying on NOC data alone is more appropriate when no *a priori* assumption can be done about the distribution of the out-of-class data (Tax et al., 1999). However, a compliant approach could be still beneficial, specifically when the highest complexity considered in SPAfPM is found in the data, meaning when all three the characteristics in Figure 7.2 are detected. In such a case, SPAfPM is called to discriminate between DKPLS and KDE-CVA. DKPLS combines dynamic and nonlinear transformations of the data prior to modeling. On the other hand, KDE-CVA employs only the dynamic transformation prior to modeling, while the nonlinear components enters at the monitoring statistics level. This is basically equivalent to applying a linear filter to data (the CVA model) and only then adopting a nonlinear approach to fault detection (KDE-based control limits). Even if the Type I error rates of the two approaches could be very similar, no clear guideline exists to determine which method will yield the best performance in terms of Type II error rate.

[13] A possible solution is to combine validation performances on all the available faulty datasets by some aggregation rule, even though this is known to be a hard task and represents a problem, for example, in selection of multi-class classification models.

Furthermore, preliminary tests of the two approaches highlighted that the parameters of the kernel functions (for example the width of the Gaussian kernel) have a dramatic effect on the Type II error rate (much more relevant than the effect they have on the Type I error), especially when the kernel transformation is applied to the data prior to modeling. This behavior is justified by the fact that kernel methods of the DKPLS kind are extremely flexible (Tax et al., 1999) and can fit the training data almost perfectly (Jia et al., 2016; Schölkopf et al., 1998).
In light of the discussion developed here, SPAfPM recommends both DKPLS and KDE-CVA as a conservative approach if all three characteristics in Figure 7.2 are detected. In this case, we also recommend assessing the distribution of the values of the performance index obtained in the rigorous cross-validation procedure, which can offer precious insights on the behavior of the models. A compliant model selection mechanism for SPAfPM is matter of future research.

7.5.5 Computational cost of model selection

Cross-validation is a computationally intensive model selection method due to its principle, based on data resampling. Specifically, the repeated k-fold scheme requires the calibration and application of rk models, while the growing windows method entails the calibration and evaluation of $k-1$ models. With the default values of r and k used in SPAfPM, 50 and 4 models must be calibrated and evaluated for static and dynamic data, respectively, for each combination of hyperparameters considered in each candidate model. This could imply a significant computational time for model selection, especially for high complexity models with several parameters and calibration procedures based on numerical optimization.
As mentioned in Section 7.2, the model library of SPAfPM includes methods with high computational efficiency in both calibration and evaluation, therefore cross-validation requires a reasonable time. The models entailing the largest computational burden in calibration are KPCA and SVDD, the complexity of which scales with N^2 due to the construction of the kernel matrix. However, even with repeated k-fold cross validation and an exhaustive grid-search scheme, the computational time for model selection does not exceed some hours for such models on a standard workstation (Dell Precision 7550 with 8-core Intel i7-10875 @ 2.3 GHz and 64GB of RAM DDR4 @ 2.933MHz, model selection run in serial mode). In the case studies described in the next Section, the most demanding model selection is the one for KPCA in the continuous carousel simulator (see Section 7.6.3), where the calibration dataset includes $N = 1260$ observations, which took around 12 hours. In all the other case studies, KPCA model selection is way less demanding, requiring seconds or minutes (and never exceeding 2 hours).
As model selection is to be performed only at the time of model comparison and, possibly, when updating the fault detection system, we believe this computational load to be fair. We also remark that the current implementation of SPAfPM is based on functions available in either standard software packages or in the previously published SPA code (Sun, 2020b) run in serial mode. Such functions are meant to be generally applicable and can undergo a substantial

optimization to reduce the computational time: future versions of SPAfPM will implement such operation. Furthermore, alternative, more computationally efficient model selection methods are matter of future research. A prominent example is represented by information criteria (Burnham et al., 2002; McQuarrie et al., 1998).

7.6 Results and discussion

This Section illustrates the effectiveness of the proposed approach on several case studies. We design a simple linear, static dataset to test SPAfPM on a trivial case. The Tennessee Eastman Process (TEP), a widely used benchmark simulator in the fault detection and diagnosis literature, is considered next. We then use a realistic simulation of a complex process, the continuous filtration and dying of paracetamol. Finally, industrial data from a metal etching process are used. For all the case studies, we consider both cases with and without quality variables. All the computations are performed in Python 3.9.12 (Python Software Foundation, 2022) and R 4.2.0 (R Foundation, 2022). The two environments are interfaced by means of rpy2 (rpy2, 2022).

7.6.1 Simulated linear dataset

A simple numerical example is designed to test the proposed framework in a controlled environment. NOC data are generated by sampling a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $V_X = 15$ variables. The mean vector, $\boldsymbol{\mu} \in \mathbb{R}^{V_X}$, is sampled from a multivariate uniform distribution $\mathcal{U}(-100 \cdot \mathbf{1}_{V_X}, 100 \cdot \mathbf{1}_{V_X})$, where $\mathbf{1}_{V_X} \in \mathbb{R}^{V_X}$ is a vector and all its components are equal to 1, hence $\mu_v \sim \mathcal{U}(-100, 100)\ \forall\ v \in \{1, \dots, 15\}$. The covariance matrix, $\boldsymbol{\Sigma} \in \mathbb{R}^{V_X} \times \mathbb{R}^{V_X}$, is generated according to the algorithm proposed by Davies et al. (2000): the 3 major eigenvalues are set to $\{7, 4, 3\}$, while the remaining 12 are drawn from uniform distributions $\mathcal{U}(0, 1/(V_X - 3))$ and constrained to sum up to 1. $N = 600$ observations are sampled from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and used as NOC data.

An additional variable is included in the dataset, to be used as quality variable to also test a case in which quality-relevant monitoring is required. The quality variable is computed as a linear combination of the V_X variables in the "process" dataset generated as described above. Combination coefficients are drawn as random real numbers from a uniform distribution $\mathcal{U}(-11, 11)$. The quality variable is summed to Gaussian noise with zero mean and variance selected so that the signal-to-noise ratio is $1{:}\,0.05$.

Three faulty datasets to be used for testing are designed as follows.

1. All the components of the mean vector are multiplied by 1.05, while the covariance matrix is left unchanged. Therefore, $\boldsymbol{\mu}_{\mathrm{F1}} = 1.05\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_{\mathrm{F1}} = \boldsymbol{\Sigma}$. 200 observations are drawn from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $N_{\mathrm{F}} = 1000$ more observations are drawn from $\mathcal{N}(\boldsymbol{\mu}_{\mathrm{F1}}, \boldsymbol{\Sigma}_{\mathrm{F1}})$, thus the fault kicks in after 200 NOC observations.

2. A new covariance matrix $\mathbf{\Sigma}_{\text{new}}$ is generated according the approach proposed by Davies et al. (2000) setting the 7 major eigenvalues to $\{3, 2.5, 2.2, 2, 1.5, 1, 0.8\}$, while the remaining 8 are drawn from uniform distributions $\mathcal{U}(0, 1/(V_X - 7))$ and constrained to sum up to 1; the mean vector is left unchanged. Therefore, $\mathbf{\mu}_{\text{F2}} = \mathbf{\mu}$ and $\mathbf{\Sigma}_{\text{F2}} = \mathbf{\Sigma}_{\text{new}}$. 200 observations are drawn from $\mathcal{N}(\mathbf{\mu}, \mathbf{\Sigma})$, then $N_{\text{F}} = 1000$ more observations are drawn from $\mathcal{N}(\mathbf{\mu}_{\text{F2}}, \mathbf{\Sigma}_{\text{F2}})$, thus the fault kicks in after 200 NOC observations.
3. The variables are replaced by dynamic variables generated as outputs of a random state-space model. The "original" variables are treated as inputs to the state-space model. The state order is an integer between 1 and 11 (inclusive) selected randomly with uniform probability. 200 burn-in observations are fed to the state-space model to guarantee that the dynamic variables are stationary processes (see Section 7.4.3 for details on the transformation mechanism). Dynamic variables are summed to independent Gaussian noise variables with zero mean and variances selected so that the signal-to-noise ratio of each dynamic variable is $1:0.1$. The dynamic variables are then scaled for their variances to match the ones of variables in the NOC dataset. 200 observations are drawn from $\mathcal{N}(\mathbf{\mu}, \mathbf{\Sigma})$, then $N_{\text{F}} = 1000$ more observations are drawn from the same distribution and transformed as explained, thus the fault kicks in after 200 NOC observations.

The calibration dataset for the proposed smart data analytics framework consists only of the NOC data and do not include any faulty data. The NOC data is first analyzed to determine the relevant data characteristics, which are used to pre-select suitable fault detection methods. Then, the candidate models are evaluated by cross-validation to tune their hyperparameters and to select the best performing one. The dataset containing the faults are treated as testing data to evaluate the rates of both Type I error (NOC observation incorrectly deemed faulty) and Type II error (faulty observations incorrectly deemed NOC). A fault is detected whenever any of the statistics of the relevant model crosses the associated control limit, coherently with the cross-validation procedure elucidated in Section 7.5.1.

The criteria introduced in Section 7.4 are used to characterize the NOC dataset available for model calibration (note that the NOC dataset is the same regardless of the presence of dependent variables in this case study). The results are the following.

- The Royston test is selected to assess non-normality. The dataset is deemed normal with a p-value of 0.7537. The dataset is deemed normal also by all the non-selected tests.
- According to the "variables" criterion, the dataset is deemed linear with a fraction of variables involved in nonlinear relationships equal to 0. All approaches to deflate the maximal correlation coefficient yield the same result.
- According to the "variables" criterion, the dataset is deemed static with a fraction of dynamic variables equal to 0.

A linear and static method is appropriate to model the NOC data. We consider cases without and with dependent variables. When no dependent variable is used, the proposed framework

selects PCA according to Figure 7.2. Hyperparameters are determined by repeated k-fold cross-validation, which yields $A = 3$ PCs. If dependent variables are considered, PLS is recommended as the most suitable model. The cross-validation procedure concludes that $A = 3$ LVs should be used. The monitoring statistics of both models applied to the testing dataset for fault 1 are shown in Figure 7.14.

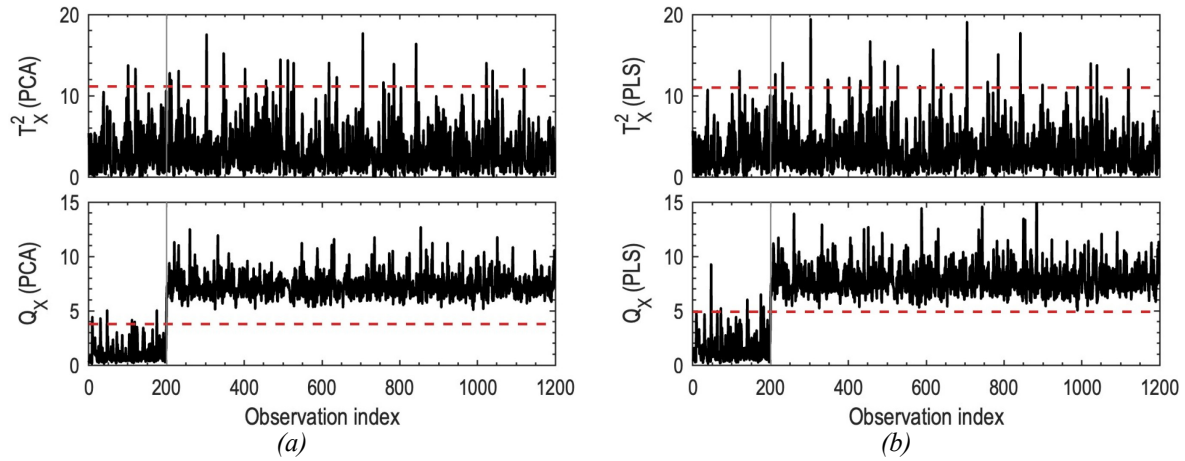


Figure 7.14. *Linear case study. Fault detection statistics of (a) PCA and (b) PLS applied to fault 1. The dashed lines represent the control limits based on the χ^2 approach. The fault occurs at observation 200 (vertical lines).*

The fault detection statistics of PCA on fault 1 are shown in Figure 7.14(a). PCA achieves a Type I error rate of 0.040 and a Type II error rate of 0.000 for fault 1 in the case where dependent variables are not used. An overview of the performance of all models not considering dependent variables in detection of all three faults considered is shown in Table 7.2. The performance of the recommended model is overall very good. The method has strong performance in terms of Type I error rate on unseen data and is the best model for Type II error rate on all faults.

The fault detection statistics of PLS on fault 1 are shown in Figure 7.14(b). In the case where dependent variables are used, PLS results in a Type I error rate of 0.035 and a Type II error

Table 7.2. *Linear case study. Overview of the Type I and Type II error rates for all methods not considering dependent variables applied to faults 1, 2, and 3.*

Fault no.		PCA	DPCA	KPCA	DKPCA	SVDD
1	Type I error rate	0.040	0.020	0.000	0.000	0.005
	Type II error rate	0.000	0.913	1.000	1.000	0.983
2	Type I error rate	0.020	0.000	0.000	0.015	0.010
	Type II error rate	0.042	0.999	1.000	1.000	0.999
3	Type I error rate	0.020	0.025	0.000	0.010	0.005
	Type II error rate	0.130	0.950	1.000	1.000	0.996

rate of 0.000 for fault 1. An overview of the performance of all models considering dependent variables in detection of the three faults considered is shown in Table 7.3. The performance of the recommended model is overall very good, with strong performance for both the Type I and II error rates. CVA and KDE-CVA achieve similar performance to PLS.

Table 7.3. *Linear case study. Overview of the Type I and Type II error rates for all methods considering dependent variables applied to faults 1, 2, and 3.*

Fault no.		PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
1	Type I error rate	0.035	0.025	0.010	0.005	0.035	0.030
	Type II error rate	0.000	0.930	0.999	1.000	0.000	0.000
2	Type I error rate	0.030	0.015	0.005	0.000	0.010	0.010
	Type II error rate	0.047	0.995	1.000	1.000	0.000	0.000
3	Type I error rate	0.030	0.020	0.000	0.005	0.010	0.010
	Type II error rate	0.121	0.956	1.000	1.000	0.003	0.003

7.6.2 Tennessee Eastman Process

The TEP is a well-known benchmark for process monitoring applications. Many different methods have been tested on the TEP to evaluate their performance in fault detection scenarios (Dong et al., 2018a; Jia et al., 2016; Ku et al., 1995; Odiowei et al., 2010; Raich et al., 1996; Rato et al., 2013; Russell et al., 2000; Samuel et al., 2016; Tien et al., 2004; Wang et al., 2014; Yin et al., 2011; Zhang et al., 2020; Zhu, 2021). The simulator has been developed by the Eastman Chemical Company to represent a real industrial chemical process consisting of a reactor, a condenser, a compressor, a separator, and a stripper (Downs et al., 1993).

The dataset generated by Chiang et al. (2001) has been selected for use in this study. NOC data, including $N = 500$ observations of the 52 process variables, were obtained by Chiang et al. (2001) running the simulator in normal operating conditions (no faults acting on the process). They performed 21 additional simulations, one for each of the 21 faults pre-implemented in the simulator, to obtain 21 faulty datasets for testing. All the faulty datasets consist of 160 observations of NOC and $N_F = 800$ additional observations of faulty conditions, which can be the result of different changes in the process. The detection difficulty of the faults varies significantly, and it is known that certain models work well on some faults, but not on others (Russell et al., 2000). Additionally, faults 3, 9, and 15 are known to be undetectable by data-driven methods (Chiang et al., 2001) and are therefore not considered in the analysis carried out herein. As for the previous case study, the data from faulty operation are used for testing to compute both the Type I and Type II error rates.

Again, we will consider both possible cases in terms of dependent variables. Out of the 52 variables featured by the TEP, 11 are manipulated variables and 41 are process measurements (Downs et al., 1993). When models not considering dependent variables (such as PCA) are

applied to the TEP, all the variables are typically included in the dataset (Chiang et al., 2001). When dependent variables are accounted for, different variables can be designated as input or dependent variables. However, most literature studies agree to consider the 11 manipulated variables and the first 21 process measurements as inputs (Jia et al., 2016; Jiao et al., 2015; Oliveri et al., 2014). There are different options for the dependent variables: for this case study we defined the mole percentage of component G in stream 9 as the dependent variable (Jiao et al., 2015; Oliveri et al., 2014). Refer to Downs et al. (1993) for details on the process/variables. The criteria introduced in Section 7.4 are used to characterize the NOC dataset available for model calibration (note that the NOC dataset differs in the cases with and without dependent variables in this case study). The results are the following.

- The Royston test is selected to assess non-normality. The dataset is deemed non-normal with a p-value basically equal to 0. This result is due to the marginal distributions of some variables. For example, variables 37 to 41 show "staircase" profiles due to their lower sampling frequency.
- According to the "variables" criterion, the dataset is deemed linear with a fraction of variables involved in nonlinear relationships equal to 0 for both cases without and with dependent variables. Such a result is in accordance with the literature (Sun, 2020a). This result, combined with the non-normality detection criterion, also allows us to conjecture that variables are either uncorrelated or only linearly correlated. An inspection of the maximal correlation and linear correlation matrices reveals that variables are mostly uncorrelated, with few cases of linear correlation (due to linear constraints among variables imposed by material balances of the process). All approaches to deflate the maximal correlation coefficient yield the same results in this case.
- According to the "variables" criterion, the dataset is deemed dynamic with a fraction of dynamic variables equal to 0.788 (41 dynamic variables out of 52) in the case without dependent variables, and equal to 0.667 (22 dynamic variables out of 33) when dependent variables are considered. Also this agrees with the literature (Sun, 2020a).

The preliminary data interrogation indicates that the NOC dataset is linear and dynamic. Considering the case without dependent variables, the data analytics triangle of SPAfPM (Figure 7.2) suggests DPCA as the best model. Growing window cross-validation is used to determine the hyperparameters of DPCA, yielding one lagged measurement ($L = 1$) and one PC ($A = 1$). In the case with dependent variables, DPLS and CVA are recommended. SPAfPM uses the growing window cross-validation to tune the hyperparameter of both models and to designate the best one between the resulting models, as described in Section 7.5.3. Figure 7.15 shows the distributions of the Type I error rate in cross-validation for the optimal DPLS and CVA models. As the average Type I errors rate in cross-validation of CVA is closer to $\alpha = 0.01$ than the one of DPLS, SPAfPM selects CVA as final model, with $L = H = 1$ and $A = 1$ as hyperparameters.

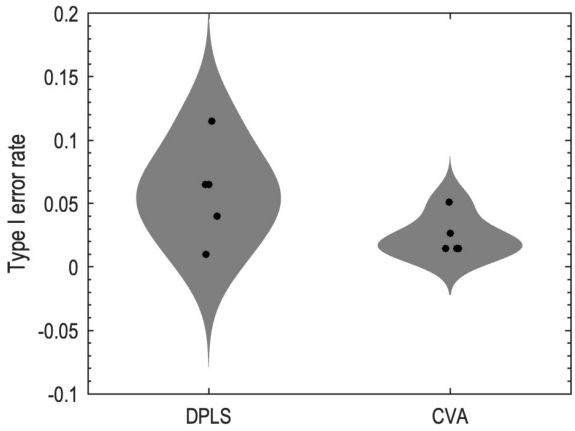


Figure 7.15. *TEP case study. Distributions of the Type I errors in cross-validation for DPLS and CVA. Each dot represents the error for one of the cross-validation blocks.*

In the following, we discuss in detail two of the available faulty datasets (faults 1 and 5). We then give an overview of the performance of SPAfPM on all available faults.

Fault 1 is one of the most frequently analyzed ones. In this case, the ratio of components A and C in stream 4 undergoes a step change, with component C increasing and component A decreasing (Downs et al., 1993). Figure 7.16 reports the fault detection statistics of DPCA and CVA for fault 1.

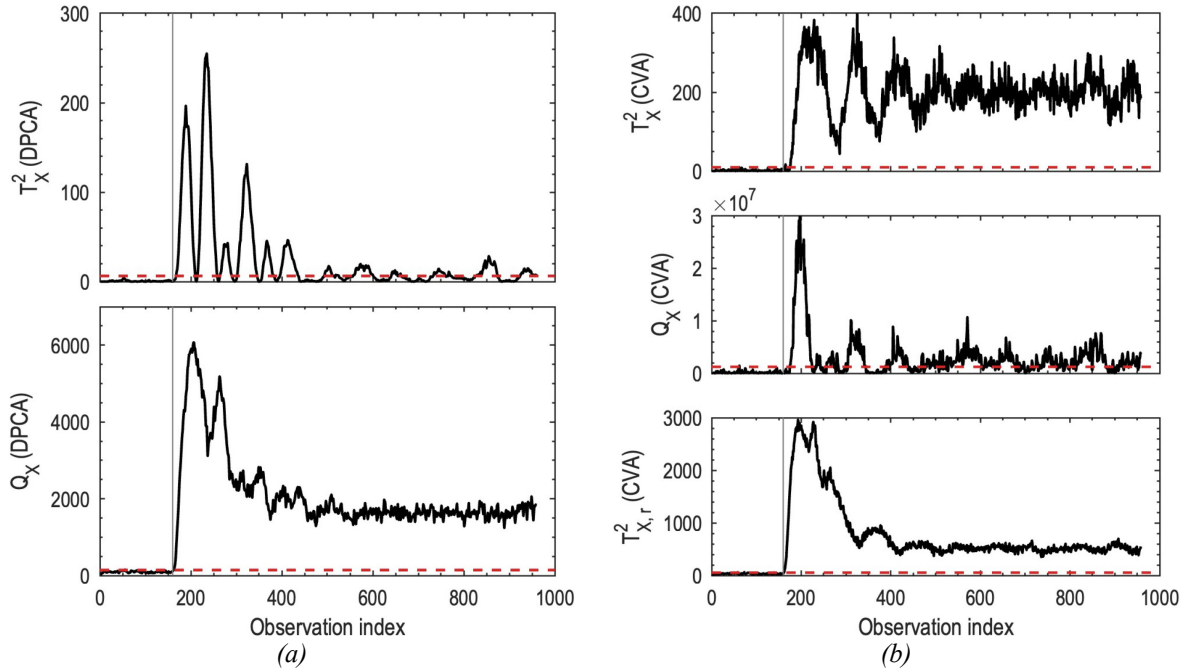


Figure 7.16. *TEP case study. Fault detection statistics of (a) DPCA and (b) CVA applied to fault 1. The dashed lines represent the control limits based on the χ^2 approach. The fault occurs at observation 160 (vertical lines).*

Table 7.4 reports an overview of the performance of all models not considering dependent variables in detection of fault 1. In terms of the Type I error rate, the suggested method, DPCA, is the second-best performing model, with a Type I error rate of 0.006, and is the best

performing model in terms of Type II error, with a rate of 0.001. Figure 7.16(a) shows that the T_X^2 statistic of DPCA crosses the control limit after the fault occurs but does not consistently detect it. On the other hand, the Q_X statistic continuously detects the fault, showing good performance for the normal operating conditions (first 160 observation) as well.

Table 7.4. *TEP case study. Overview of the Type I and Type II error rates for all methods not considering dependent variables applied to fault 1.*

	PCA	DPCA	KPCA	DKPCA	SVDD
Type I error rate	0.019	0.006	0.006	0.000	0.025
Type II error rate	0.004	0.001	1.000	0.098	0.003

Moving to the case where dependent variables are considered separately, an overview of the performance of the models in terms of Type I error rate and Type II error rate is shown in Table 7.5. CVA has relatively high Type I error rate, but consistently detects the fault, resulting in a Type II error rate of 0.000. Figure 7.16(b) highlights that both the T_X^2 and $T_{X,\mathrm{r}}^2$ statistics detect the fault perfectly. However, the latter is quite sensitive, leading to a high Type I error rate. Note that DPLS (the alternative method suggested by SPAfPM) performs well too in terms of both Type I and Type II error rate, even though not as well as CVA (see Table 7.5).

Table 7.5. *TEP case study. Overview of the Type I and Type II error rates for all methods considering dependent variables applied to fault 1.*

	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
Type I error rate	0.013	0.031	0.000	0.019	0.050	0.000
Type II error rate	0.001	0.000	0.008	0.006	0.000	0.003

Fault 5 is also commonly used to evaluate fault detection methods. In this case, the inlet temperature of the condenser cooling water experiences a step change, affecting the reactor cooling water flow rate (Downs et al., 1993). Figure 7.17 reports the fault detection statistics of DPCA and CVA applied to fault 5.

The performance in detection of fault 5 by methods not considering dependent variables are shown in Table 7.6. DPCA shows a higher Type I error rate compared to other methods; however, it is the second-best option for the Type II error rate (after SVDD). The Type II error rate of DPCA is still high. The reason for this result is that methods that do not consider dependent variables are not suitable to consistently detect fault 5 (Chiang et al., 2001). This is made clear by Figure 7.17(a), which shows that the fault detection statistics of DPCA can detect the fault after its onset but return below their control limits shortly thereafter.

If dependent variables are considered separately, our algorithm recommends CVA. CVA yields a Type I error rate of 0.044, comparing well to other methods, as shown in Table 7.7. In terms

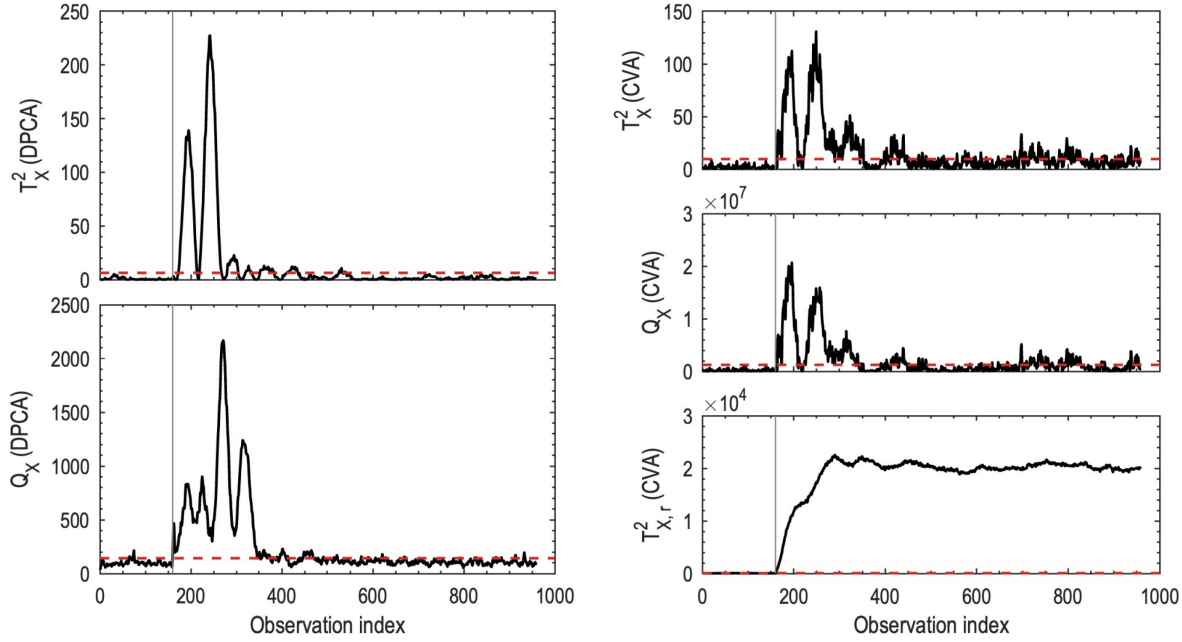


Figure 7.17. *TEP case study. Fault detection statistics of (a) DPCA and (b) CVA applied to fault 5. The dashed lines represent the control limits based on the χ^2 approach. The fault occurs at observation 160 (vertical lines).*

Table 7.6. *TEP case study. Overview of the Type I and Type II error rates for all methods not considering dependent variables applied to fault 5.*

	PCA	DPCA	KPCA	DKPCA	SVDD
Type I error rate	0.038	0.056	0.019	0.031	0.038
Type II error rate	0.606	0.578	0.927	0.630	0.551

of Type II error rate, CVA clearly outperforms the other methods and achieves a flawless Type II error rate of 0.000 (KDE-CVA achieves similar performance). Figure 7.17(b) illustrates how only the $T^2_{X,\mathrm{r}}$ statistic is capable of consistently detecting the fault after its occurrence. Being such a statistic unique to CVA (and KDE-CVA), this motivates why this method achieves such a good performance on fault 5. This case further proves that the recommendation of CVA made by SPAfPM is appropriate.

Table 7.7. *TEP case study. Overview of the Type I and Type II error rates for all methods considering dependent variables applied to fault 5.*

	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
Type I error rate	0.038	0.069	0.000	0.025	0.044	0.000
Type II error rate	0.616	0.565	0.702	0.639	0.000	0.000

The proposed framework suggests DPCA and CVA in the cases without and with dependent variables, respectively, solely relying on NOC data. However, 21 different faults can occur in

the TEP process (18 of which are undetectable by data-driven methods), and it is known that there is no single model that achieves flawless detection performance on all faults (Russell et al., 2000). We are nonetheless interested in evaluating how the selected models compare to the other models on all the 18 detectable faults in terms of both Type I and Type II error rates.
All models show similar performance regarding the Type I error rate, as expected, with kernel-based methods tending to perform slightly better on average (not shown for brevity). Concerning the Type II error rate, Table 7.8 shows an overview of the performance of all models not considering dependent variables: DPCA is observed to be the best performing model in 12 out of 18 cases, with performance almost identical to the best ones (achieved by SVDD) in the remaining 6 cases. Similarly, Table 7.9 shows an overview of the Type II error rates for all models considering dependent variables: CVA is the best performing model in 16 out of 18 cases; DPLS, the alternative model suggested by SPAfPM, is the best model in the remaining 2 cases. These results demonstrate the effectiveness and strong performance of the proposed smart data analytics approach for selecting the best method for fault detection.

Table 7.8. *TEP case study. Overview of the Type II error rates for all methods not considering dependent variables applied to all detectable faults. Error rates of the best performing method for each fault are highlighted in bold font.*

Fault no.	PCA	DPCA	KPCA	DKPCA	SVDD
1	0.004	**0.001**	1.000	0.098	0.003
2	0.014	**0.013**	0.986	0.936	0.014
4	0.063	**0.009**	0.935	0.838	0.029
5	0.606	0.578	0.927	0.630	**0.551**
6	0.000	**0.000**	1.000	0.966	0.000
7	0.000	**0.000**	0.399	0.338	0.000
8	0.013	0.011	0.899	0.117	**0.006**
10	0.345	0.320	0.770	0.404	**0.316**
11	0.273	**0.207**	0.849	0.721	0.228
12	0.006	**0.003**	0.896	0.142	0.004
13	0.040	**0.036**	0.990	0.467	0.041
14	0.000	**0.000**	0.957	0.058	0.000
16	0.419	**0.350**	0.862	0.484	0.352
17	0.068	**0.058**	0.966	0.576	0.064
18	0.086	0.084	0.995	0.934	**0.075**
19	0.799	**0.737**	0.972	0.906	0.738
20	0.333	0.292	0.931	0.475	**0.273**
21	0.539	0.503	0.947	0.619	**0.500**

7.6.3 Continuous filtration and drying of paracetamol

Destro et al. (2021) developed a detailed, highly nonlinear, and mechanistic model of a process for continuous filtration and drying of an active pharmaceutical ingredient (paracetamol): the continuous carousel simulator (ContCarSim; Destro et al., 2022a). An open-source simulator implementing such model is freely available on GitHub (Destro et al., 2022b). The process is carried out in a revolving carousel unit with five slots, the so-called "ports." A slurry containing the crystals, the mother liquor, and the solvent is loaded to the first port; vacuum-driven de-liquoring takes place in the second and third ports; the fourth port is used for drying the crystals under a flow of hot air; the cake of dry crystal is discharged from the fifth port. Fouling of the filter meshes is also simulated, and an automatic cleaning routine is implemented by the simulator. Measurements from $V_X = 8$ sensors installed on the actual machine used for model development are returned by the simulator. The reader is referred to the original literature resources for details on the model (Destro et al., 2021) and on the simulator (Destro et al., 2022a).

Table 7.9. *TEP case study. Overview of the Type II error rates for all methods considering dependent variables applied to all detectable faults. Error rates of the best performing method for each fault are highlighted in bold font.*

Fault no.	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
1	0.001	0.000	0.008	0.006	**0.000**	0.003
2	0.015	0.013	0.021	0.018	**0.008**	0.014
4	0.019	0.000	0.835	0.809	**0.000**	0.000
5	0.616	0.565	0.702	0.639	**0.000**	0.000
6	0.000	0.000	0.004	0.004	**0.000**	0.000
7	0.000	0.000	0.029	0.000	**0.000**	0.000
8	0.010	0.010	0.035	0.025	**0.010**	0.021
10	0.380	0.294	0.514	0.424	**0.073**	0.111
11	0.261	0.140	0.615	0.576	**0.127**	0.235
12	0.006	0.000	0.024	0.006	**0.000**	0.000
13	0.045	**0.036**	0.056	0.054	0.043	0.048
14	0.000	0.000	0.005	0.000	**0.000**	0.000
16	0.439	0.296	0.674	0.487	**0.041**	0.082
17	0.061	0.040	0.211	0.157	**0.025**	0.039
18	0.080	**0.073**	0.098	0.088	0.082	0.098
19	0.806	0.622	0.986	0.950	**0.044**	0.097
20	0.365	0.276	0.583	0.474	**0.083**	0.092
21	0.514	0.427	0.620	0.496	**0.302**	0.391

ContCarSim is used as a case study to test the proposed smart process analytics framework. MATLAB R2022a (The Mathworks, 2022a) is used to run the simulator. All the simulations are performed setting ContCarSim to "open-loop mode" with sampling time of 1 second. As some of the simulated sensor measurements returned by ContCarSim are noise-free, the code of the simulator is slightly modified before data generation:

- the variance of the measurement error affecting the inlet slurry concentration sensor, AI101, is set to 4;
- Gaussian noise with zero mean and standard deviation equal to $4 \cdot 10^{-5}$ is added to the slurry level sensor, LI101;
- Gaussian noise with zero mean and standard deviation equal to $2.4 \cdot 10^{-8}$ is added to the slurry volume sensor, VI101;
- Gaussian noise with zero mean and standard deviation equal to 416.7 is added to the inlet and outlet pressure sensors on the second and third ports, PI101 and PI102, respectively;
- Gaussian noise with zero mean and standard deviation equal to 0.3334 is added to the inlet and outlet drying air temperature sensors on the fourth port, TI101 and TI102, respectively, before the rounding performed by the simulator;
- Gaussian noise with zero mean and standard deviation equal to 0.03334 is added to the drying air flow rate, FI101, before the rounding performed by the simulator.

With the above settings, NOC data are generated running the simulator with integration time set to 1950 seconds, which yields 1950 observations. Time profiles returned are processed to be used for calibration of fault detection models. ContCarSim performed a realistic simulation of the process, therefore recording measurements on ports "in real time". This means that the measurements from the drying port feature a delay of three cycle-times, which is the time between when the slurry is fed to the first port and when the cake reaches the fourth port. Such a delay is corrected as suggested by Kourti et al. (1995): by shifting measurements of sensors on the fourth port (TI101, TI102, and FI101) by the appropriate time. After data shift, three more processing operations are performed:

- the simulator saves initial conditions of all variables after each port shift: these are removed in order to avoid having different measurements at the same time instant;
- the parts of the time profiles concerning cleaning phases are discarded, as only the actual filtering-drying process is of interest for the case study;
- only observations of cakes that have undergone a complete cycle (passed through all five ports) are retained, therefore data of incomplete cycles are discarded.

The data processing operations yield the final NOC dataset, including $N = 1260$ observations. Sensor data collected in the dataset concern the process only. For the purpose of testing also the performance of quality-relevant monitoring, one of the simulation states is selected as quality variable to characterize the product: the solvent concentration in the cake being

processed. Gaussian noise with zero mean and standard deviation equal to 0.0008 is added to such a state to simulate noise of a real measurement. Finally, data of the output variables undergo the same processing applied to the process data, as to guarantee time alignment.

ContCarSim provides two pre-implemented faults, called "disturbance scenarios" in the simulator. Scenario number 1, a ramp change in the feed slurry concentration, is selected to generate the faulty dataset. To generate the testing dataset, scenario 1 is enabled, and the simulation is run with integration time set to 2490 seconds. Faulty data are processed in the same way as the NOC ones, which yields a dataset containing 1620 observations. The simulator is implemented in such a way that the fault onsets at simulation time $t = 301$ seconds (considering also cleaning operations). Therefore, the first 210 observations (after data processing) of the faulty dataset are NOC, while the actual faulty observations are $N_F = 1410$. The criteria introduced in Section 7.4 are used to characterize the NOC dataset available for model calibration (note that the NOC dataset is the same regardless of the presence of dependent variables in this case study). The results are the following.

- The Royston test is selected to assess non-normality. The dataset is deemed non-normal with a p-value basically equal to 0. The dataset is deemed non-normal also by all the non-selected tests.
- According to the "variables" criterion, the dataset is deemed nonlinear with a fraction of variables involved in nonlinear relationships equal to 0.625 (5 variables out of 8). This result is expected due to the high nonlinearity of the process model and to the absence of a control system (the simulator is run in "open-loop mode"). All approaches to deflate the maximal correlation coefficient yield the same result.
- According to the "variables" criterion, the dataset is deemed dynamic with a fraction of dynamic variables equal to 0.625 (5 dynamic variables out of 8).

The above results imply that nonlinearity and dynamics need to be accounted for when building the model of the NOC data. Similarly to the previous case study, we consider cases without and with dependent variables. In the former scenario, SPAfPM selects DKPCA according to Figure 7.2. Growing window cross-validation is used to determine the hyperparameters, which turn out to be $L = 1$ lags, Gaussian kernel with $\sigma = 50$, and $A = 1$ PC. In the case where dependent variables are considered, DKPLS and KDE-CVA are selected by SPAfPM as candidate models, optimized by cross-validation, and compared to identify the best model. Based on the error distributions reported in Figure 7.18, KDE-CVA achieves an average deviation of the Type I error rate from $\alpha = 0.01$ in cross-validation lower than the one of DKPLS (also the variance of the error is lower), therefore it is designated as the most suitable model. The hyperparameter tuning for KDE-CVA yields $L = H = 3$, $A = 3$, $\xi_{T_X^2} = \xi_{T_{X,\mathrm{r}}^2} = 2$, and $\xi_{Q_X} = 5$. However, DKPLS is considered as a possible alternative model (recall the discussion in Section 7.5.4); hyperparameters for DKPLS result in $L = 1$, $k_\mathrm{p} = k_\mathrm{rbf}$ with $\sigma = 50$, and $A = 1$. The fault detection statistics of the two selected methods applied to fault 1 are reported in Figure 7.19.

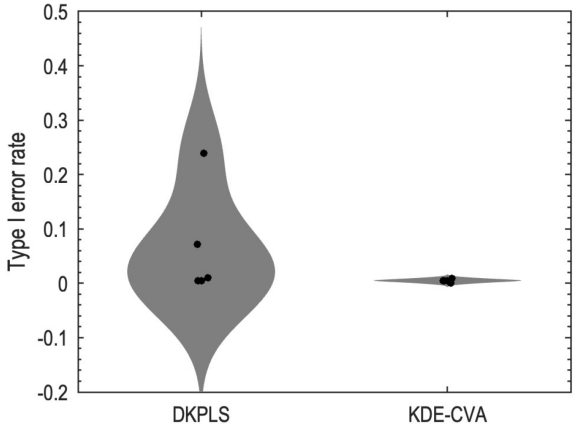


Figure 7.18. *ContCarSim case study. Distributions of the Type I errors in cross-validation for DKPLS and KDE-CVA. Each dot represents the error for one of the cross-validation blocks.*

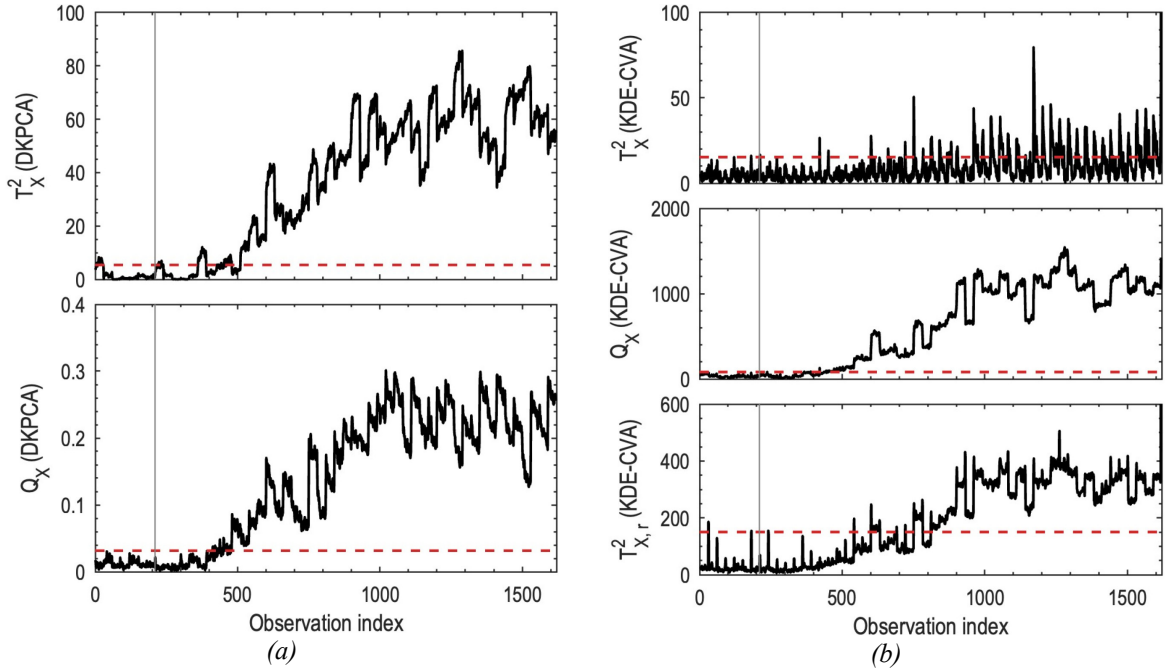


Figure 7.19. *ContCarSim case study. Fault detection statistics of (a) DKPCA and (b) KDE-CVA applied to fault 1. The dashed lines represent the control limits based on the χ^2 approach. The fault occurs at observation 210 (vertical lines).*

When applied to fault 1, DKPCA for the case without dependent variables results in a Type I error rate of 0.100 and a Type II error rate of 0.124. An overview of the performance of all models not considering dependent variables in detection of fault 1 is reported in Table 7.10. DKPCA is the second-best performing model for the Type II error (being fundamentally equivalent to the best model, DPCA) and performs well for the Type I error rate too. Only SVDD seems to perform significantly better for the Type I error rate.

If dependent variables are considered, the overall performance of the model selected by SPAfPM, KDE-CVA, in detection of fault 1 is compared to the ones of the other models considering dependent variables in Table 7.11. KDE-CVA yields reasonable Type II error and

Table 7.10. *ContCarSim case study. Overview of the Type I and Type II error rates for all methods not considering dependent variables applied to fault 1.*

	PCA	DPCA	KPCA	DKPCA	SVDD
Type I error rate	0.095	0.114	0.090	0.100	0.033
Type II error rate	0.133	0.123	0.143	0.124	0.129

shows good performance for the Type I error. The alternative model, DKPLS, performs marginally better. This example further illustrates the point discussed in Section 7.5.4. We shall also remark that the nature of the fault (a ramp increase) induces a certain detection delay in all models included in SPAfPM. Therefore, the Type II error rates obtained for all models and reported in Table 7.10 and in Table 7.11 are reasonable.

Table 7.11. *ContCarSim case study. Overview of the Type I and Type II error rates for all methods considering dependent variables applied to fault 1.*

	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
Type I error rate	0.033	0.029	0.029	0.038	0.043	0.033
Type II error rate	0.134	0.122	0.162	0.125	0.147	0.158

7.6.4 Industrial metal etching process

The last case study considered in this Chapter is based on data collected in an industrial plasma etch process for semiconductor manufacturing (Wise et al., 1999). The dataset can be freely downloaded (Eigenvector Research, Inc., 1999). Wafers are etched in a recipe-driven batch process carried out in a commercial Lam 9600 plasma etch machine. Integrated sensors perform the online measurements collected to build the dataset, which includes 19 variables (plus the timestamp of measurements and a numerical identifier of the processing phase, which are disregard in this study). Among the variables, the "RFB reflected power" and the "TCP reflected power", as named by Wise et al. (1999), are binary variables (discrete with two levels). Also, many variables appear to vary on discrete levels due to accuracy of the sensors. The complete dataset collects 108 batches under normal operating conditions. Furthermore, 21 wafers are manufactured under faulty operating conditions. For a detailed description of the process and data, refer to Wise et al. (1999).

Given the richness of features, the dataset described above quickly became a benchmark for batch process monitoring systems, with many studies considering it to test novel fault detection methods (Azamfar et al., 2020; Camacho et al., 2006a; Chen et al., 2010; Goodlin et al., 2003; He et al., 2011; Lv et al., 2018; Wang et al., 2015). In particular, it is known that this dataset features a high degree of correlation among variables (Cherry et al., 2007) and a well-defined multi-phase dynamics (Camacho et al., 2006a). Furthermore, the distribution of the data is highly non-normal (Chen et al., 2010). Given the presence of correlated variables and the non-

normality of the data, a significant fraction of variables is expected to be involved in nonlinear relationships.

For the application of SPAfPM, one single calibration batch is used: the first one, named `l2901.txm` in the `MACHINE Data.mat` dataset, which contains $N = 112$ observations. On the other hand, faults 1, 10, and 16, named "TCP +50", "TCP +30", and "TCP -15", respectively (Wise et al., 1999), are selected for testing. Three faulty datasets are obtained stacking another NOC batch (file `l2902.txm` in the `Data.mat` dataset), counting 107 observations, and the three aforementioned faults (files `l2915.txm`, `l3120.txm`, and `l3318.txm`, respectively). The faulty datasets contain 210, 207, and 207 observations, respectively, and faults onset at observation 107 in all of them.

As for the other case studies, both scenarios without and with dependent variables are assessed. In the former case, all the $V_X = 19$ variables are considered. In the latter, variable 10, the "phase error" in Wise et al. (1999), is selected as quality variable, while the remaining 18 variables are kept as process variables. The two discrete variables mentioned above are actually considered as such for the assessment of the characteristics of the calibration dataset. On the other hand, variables varying on discrete levels are considered numerical, as this pattern is due to sensor accuracy.

The criteria introduced in Section 7.4 are used to characterize the NOC dataset available for model calibration (note that the NOC dataset differs in the cases with and without dependent variables in this case study). The results are the following.

- The Royston test is selected to assess non-normality. The dataset is deemed non-normal with a p-value basically equal to 0. This result was expected due the presence of binary variables and to the fact that many variables vary on discrete levels due to measurement accuracy.
- According to the "variables" criterion, the dataset is deemed nonlinear with a fraction of variables involved in nonlinear relationships equal to 0.4737 (9 nonlinear variables out of 19) in the case without dependent variables, and equal to 0.4444 (8 nonlinear variables out of 18) in the case with dependent variables. All approaches to deflate the maximal correlation coefficient yield essentially the same results, yet with some minor variations in the fraction of nonlinear variables.
- According to the "variables" criterion, the dataset is deemed dynamic with a fraction of dynamic variables equal to 0.647 (11 dynamic variables out of 17 non-discrete variables) in the case without dependent variables, and equal to 0.625 (10 dynamic variables out of 16 non-discrete variables) when output variables are considered.

The data interrogation criteria of SPAfPM indicate that dynamics and nonlinearity are present in the NOC data. We consider cases without and with dependent variables. If all the $V_X = 19$ variables in the dataset are considered process variables, DKPCA would usually be selected according to the Figure 7.2. However, due to the presence of discrete variables, SPAfPM

recommends SVDD as the most suitable model (see discussion in Section 7.3.3). The hyperparameters of SVDD are tuned with repeated k-fold cross-validation, which results in a Gaussian kernel with $\sigma = 50$ and a radius-to-coverage parameter $C = 0.2$. If dependent variables are considered, DKPLS and KDE-CVA are recommended by SPAfPM based on the found data characteristics. Growing window cross-validation is used to first determine the hyperparameters of both models, then to select the best one. The error distributions reported in Figure 7.20 suggest that DKPLS is to be preferred, having a consistently lower deviation of the Type I error rate from $\alpha = 0.01$ in cross-validation, while the KDE-CVA yields large deviations for some of the blocks.

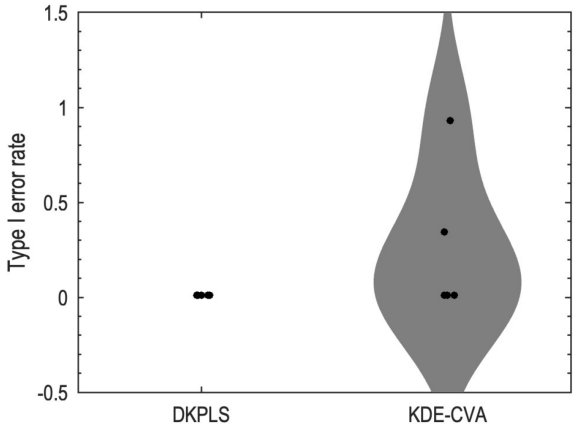


Figure 7.20. *Metal etching process case study. Distributions of the Type I errors in cross-validation for DKPLS and KDE-CVA. Each dot represents the error for one of the cross-validation blocks.*

However, note that only $N = 102$ observations are included in the calibration dataset. While this small number of observations can still be handled by the criteria for the preliminary data interrogation, it compromises the reliability of the cross-validation procedure. In particular, each data block counts only $N_{\text{val}} \simeq 20$ observation, which leads the resolution of the Type I error rate to degrade to approximately $1/20 = 0.05$. This in turn amplifies the variability of the distribution of the Type I error rate in cross-validation, as large errors are obtained even with just one or two observations beyond the control limits. These facts explain the extreme errors shown by KDE-CVA in Figure 7.20. Considering these points and based on the discussion in Section 7.5.4, SPAfPM recommends both DKPLS and KDE-CVA as potentially suitable methods. The hyperparameter tuning for DKPLS results in $L = 1$ lags, polynomial kernel with $c_0 = 1$, $d = 3$, and $\gamma = 0.0004$, and $A = 1$ LV. The hyperparameters for KDE-CVA are $L = H = 1$, $A = 1$, $\xi_{T_X^2} = \xi_{T_{X,\text{r}}^2} = 1$, and $\xi_{Q_X} = 1$.

Fault 10 is selected and discussed as example in this case study. The fault detection statistics of SVDD, DKPLS, and KDE-CVA are show in Figure 7.21.

An overview of the performance of all models not considering dependent variables in detection of all three faults considered is shown in Table 7.12. If no dependent variables are used, SVDD achieves a Type I error rate of 0.028 and a Type II error rate of 0.071 for fault 10. The

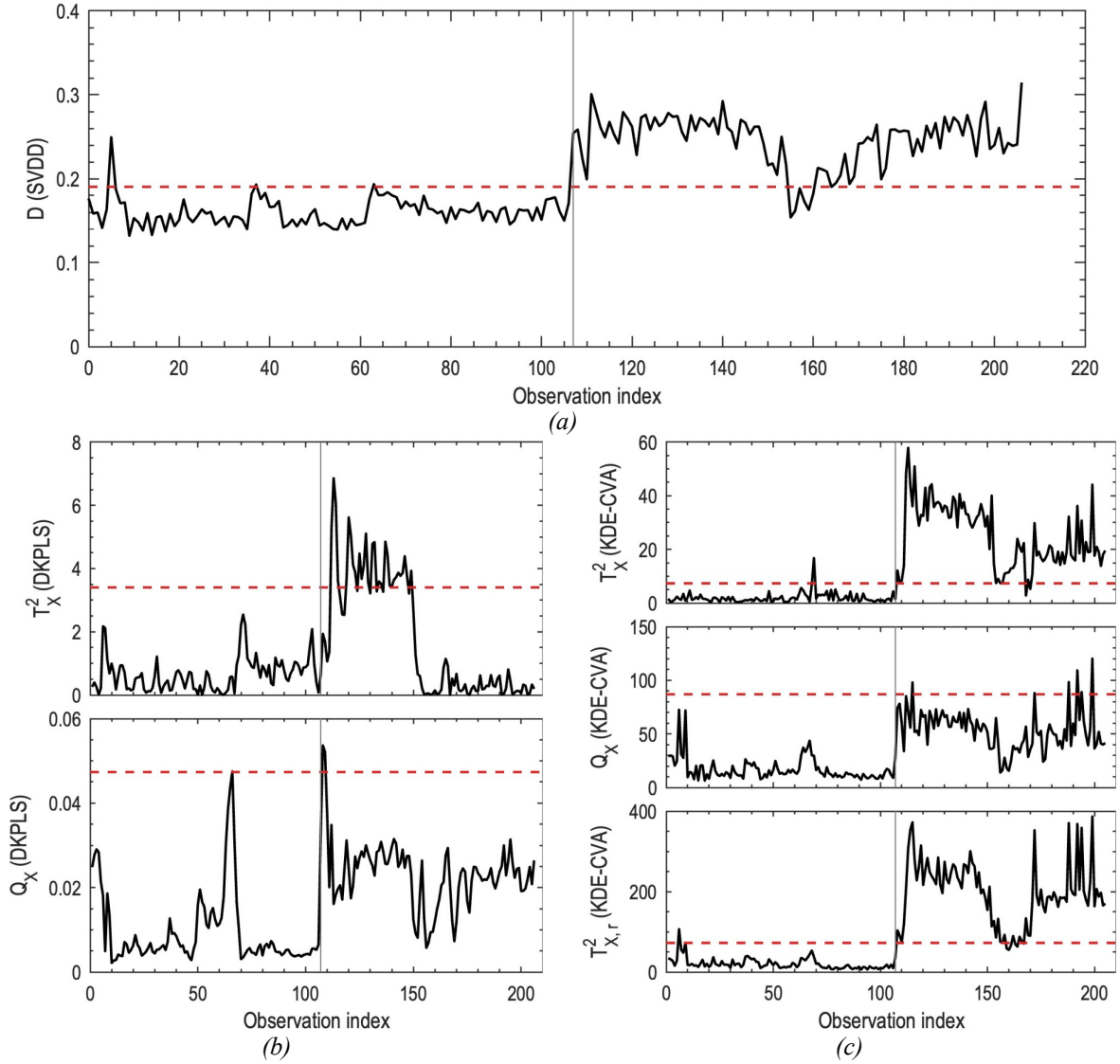


Figure 7.21. *Metal etching process case study. Fault detection statistics of (a) SVDD, (b) DKPLS, and (c) KDE-CVA applied to fault 10. The dashed lines represent the radius of the hypersphere in (a), and control limits based on the χ^2 approach in (b) and (c). The fault occurs at observation 107 (vertical lines).*

performance of the recommended model in detection of fault 10 is overall very good. SVDD shows good performance for the Type I error rate and performs significantly better than all other models for the Type II error rate. Similar results are observed for faults 1 and 16 (note that the Type I error rate is consistent over all faults as the NOC data in testing are the same). These findings further back up the decision to default the model selection to SVDD if discrete variables are found in the dataset, as discussed in Paragprah 7.3.3.

In the case where dependent variables are used, KDE-CVA outperforms DKPLS in fault 10, as can be seen in Table 7.13, which provides an overview of the performance of all models considering dependent variables in detection of all three faults considered. KDE-CVA results in a Type I error rate of 0.028 and a Type II error rate of 0.000 for fault 10. This is the best result, as equivalent to CVA. Figure 7.21(b) highlights that DKPLS detects the fault after its

Table 7.12. *Metal etching process case study. Overview of the Type I and Type II error rates for all methods not considering dependent variables applied to faults 1, 10, and 16.*

Fault no.		PCA	DPCA	KPCA	DKPCA	SVDD
1	Type I error rate	0.019	0.000	0.000	0.000	0.028
	Type II error rate	0.981	0.980	1.000	1.000	0.139
10	Type I error rate	0.019	0.009	0.000	0.000	0.028
	Type II error rate	0.600	0.354	1.000	1.000	0.071
16	Type I error rate	0.019	0.009	0.000	0.000	0.028
	Type II error rate	0.860	0.616	1.000	1.000	0.041

onset by the T_X^2 statistic; however, such statistic falls below the control limit around observation no. 150, when the intensity of the fault seems to decrease. On the other hand, the T_X^2 and $T_{X,\mathrm{r}}^2$ statistics of KDE-CVA show a higher sensitivity to the fault and stay beyond their control limits for a longer time, as seen in Figure 7.21(c), providing a consistent detection of the fault 10. The performance of KDE-CVA on other faults is overall very good. The model shows great performance on both the Type I and Type II error rates, performing second-best on fault 1 and best on faults 10 and 16.

Table 7.13. *Metal etching process case study. Overview of the Type I and Type II error rates for all methods considering dependent variables applied to faults 1, 10, and 16.*

Fault no.		PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
1	Type I error rate	0.019	0.009	0.000	0.009	0.037	0.028
	Type II error rate	0.990	0.971	1.000	0.990	0.530	0.828
10	Type I error rate	0.019	0.019	0.000	0.009	0.037	0.028
	Type II error rate	0.670	0.374	1.000	0.673	0.000	0.000
16	Type I error rate	0.019	0.019	0.000	0.019	0.037	0.028
	Type II error rate	0.830	0.616	1.000	0.867	0.000	0.000

7.7 Conclusions

We proposed a smart data analytics approach to fault detection in this Chapter. The approach is implemented by the SPAfPM software and is geared towards data from real manufacturing processes. A preliminary data interrogation allows to determine data characteristics to select a set of appropriate candidate models for fault detection on the data at hand; a rigorous model selection and discrimination procedure, implemented by means of cross-validation, then designates the best model among the candidates. These operations constitute the backbone of the automation mechanism of the proposed framework.

SPAfPM conducts a preliminary interrogation of the data at hand to search for three relevant characteristics: nonlinear correlation among variables (equivalent to non-normality of the data distribution); dynamics in the data in the form of correlation among observations; availability of variables describing the product quality. We developed criteria to automatically detect the former two characteristics and validated them in a multitude of different Monte Carlo simulations. On the other hand, the attribution of the third characteristic requires knowledge on the process that generated the data, therefore it is left as a decision to be made by the user of the software.

Candidate models able to cope with the found data characteristics are calibrated. A rigorous cross-validation procedure is used to tune their hyperparameters and, subsequently, to designate the best model for the problem at hand. To this end, we designed an index to measure the fault detection performance of different models under the very stringent assumption that only NOC data are available at the model calibration stage, as it is unlikely that a comprehensive database comprising all possible faulty operating conditions is available in the common industrial practice. The inclusion of faulty data in the model selection procedure has some benefits nonetheless, therefore it represents a path for future research.

We demonstrated the effectiveness of the proposed smart data analytics approach for fault detection on four case studies: a simulated, linear dataset; the Tennessee Eastman Process; a simulation of a continuous filtration and drying of paracetamol; an industrial metal etching process for semiconductor manufacturing. In particular, the TEP is an established benchmark for process monitoring systems. Based on the preliminary analysis of the NOC data available for such a case study, the framework selected DPCA when dependent variables were not considered separately, while CVA was recommended if dependent variables were considered as such. DPCA proved to be the best performing model among the ones not considering dependent variables on 12 out of the 18 faults used for testing (in terms of the Type II error rate); CVA was the best performing model among the ones considering dependent variables for 16 out of 18 faults. Similar results were obtained in the other case studies.

The outcomes of the case studies prove the strong performance of the proposed framework: SPAfPM successfully determined model performing best for most faults. Overall, the proposed approach successfully suggests the most appropriate model and determines the optimal set of hyperparameters based on a rigorous, fully automated procedure, all requiring minimal expert knowledge on fault detection by the user.

Conclusions and future prospects

The studies presented in this Thesis represent a step forward in the digitalization of industrial biorefineries and provide strong evidence of the value of the Industry 4.0 approach. Advanced data analytics methods are the foundation of Industry 4.0 and were leveraged to pursue the following major objectives of this Thesis.

1. Provide evidence that Industry 4.0 is a precious tool for industrial biorefineries.
2. Contribute to the methodological advancement of data-driven modeling.

The first objective was accomplished developing process understanding and digital support systems to enhance the operations of the one-of-a-kind industrial biorefinery manufacturing 1,4-butanediol by bioconversion of renewable biomass. Specifically, the data-driven process understanding of the bioconversion step in the upstream was presented in Chapter 3, while Chapter 4 and Chapter 5 discussed a comprehensive investigation on membrane fouling in the downstream section, carried out by hybrid modeling and feature-oriented modeling, respectively. The unique industrial environment considered in this Thesis and its specific modeling challenges suggested ways to improve the existing methods, thus accomplishing the second objective. A novel approach to latent-variable model inversion was proposed in Chapter 6, while Chapter 7 discussed the development of an automated framework for selection and calibration of data-driven fault detection systems. An overview of the main achievements of this Thesis is reported in Table C.1. Additional details on each Chapter are provided below.

A **comprehensive analysis of the bioconversion step** in the upstream process was presented in Chapter 3. An array of seven fed-batch bioreactors operated in cycled mode, used to manufacture the main biorefinery product, suffered from a decrease in the end-of-batch product quality. Data-driven methods were used to gain **process understanding** and to model the end-of-batch quality. Model interpretation uncovered a strong interaction among the process variables and offered a physically meaningful explanation of the potential causes of the quality loss. These conclusions were verified by latent-variable model inversion to formulate guidelines for recovering the product quality. Unfortunately, the results could not be validated experimentally due to significant changes taking place in the plant set-up (unrelated to the investigated problem) simultaneously to the completion of this study.

Chapter 4 developed a **soft sensor to estimate membrane resistances based on a hybrid modeling strategy**. The ultrafiltration unit in the downstream process is a critical operation, as it separates the biomass from the solution containing the product formed in the upstream. Seven tightly interconnected membrane modules realize such a separation, but the process is affected by severe membrane fouling. However, the current fouling monitoring strategy entails multiple drawbacks, as it relies on profiles of process variables manually acquired by operators through

Table C.1. *Summary of the achievements of this Thesis, with indication of their relevant references, organized by Chapter.*

Chapter	Application	Data origin	Main achievements	References
Chapter 3	• Bioconversion unit, upstream process	Industrial (upstream)	• Investigated differences among supposedly identical bioreactors • Identified causes of decreasing trend in product quality • Developed guidelines for recovery of process and product quality	
Chapter 4	• Ultrafiltration unit, downstream process	Industrial (downstream)	• Developed a soft sensor to monitor fouling of individual membranes • Provided evidence of the several advantages of monitoring fouling using membrane resistances • Leveraged the hybrid modeling paradigm to simplify the model	Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2024). Hybrid modeling of a biorefinery separation process for performance monitoring. *Chemical Engineering Science* **283**, 119413. https://doi.org/10.1016/j.ces.2023.119413.
Chapter 5	• Ultrafiltration unit, downstream process	Industrial (downstream)	• Confirmed effectiveness of cleaning policies in the plant • Validated plant strategies to counteract reversible fouling • Uncovered interaction between reversible and irreversible fouling • Developed guidelines to improve membrane replacement policies • Leveraged a feature-oriented modeling method to build process knowledge into the analysis and ease model interpretation	Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023d). *Understanding fouling in an industrial biorefinery membrane separation process by feature-oriented data-driven modeling* [In preparation]. Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023c). Troubleshooting high-pressure issues in an industrial biorefinery process by feature-oriented modeling. In: *Computer-Aided Chemical Engineering* **52**, *Proceedings of the 33rd European Symposium on Computer Aided Process Engineering (ESCAPE33)*, 163–168. https://doi.org/10.1016/B978-0-443-15274-0.50027-5.

Table C.1 *(continued).*

Chapter	Application	Data origin	Main achievements	Reference
Chapter 6	• Batch fermentation process • Fed-batch penicillin production process	Simulated	• Proposed novel method for latent-variable model inversion • Correlated quality variables can be used in the modeling phase • Demonstrated advantages of using all quality variables in inversion • Better uncertainty estimation on the inversion solution • Improved methods for estimation of the null space uncertainty	Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2022). Digital design of new products: accounting for output correlation via a novel algebraic formulation of the latent-variable model inversion problem. *Chemometrics and Intelligent Laboratory Systems*, **227**(June), 104610. https://doi.org/10.1016/j.chemolab.2022.104610. Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2023a). Estimation of null-space uncertainty in latent-variable model inversion: The case of correlated quality attributes. In: *XI Colloquium Chemometricum Mediterraneum – Book of Abstracts*, 144–145.
Chapter 7	• Numerical case • Tennessee Eastman Process • Continuous filtration and drying of drugs • Semiconductor manufacturing	Simulated and industrial (data from the literature)	• Proposed a framework for automatic development of data-driven fault detection models • Developed rigorous criteria to assess properties of data at hand • Designed performance indexes for fault detection to compare and evaluate candidate models • Implemented model selection procedure based on normal operating conditions data only • The framework selects the best model in all case studies • Implemented the framework as open-source software to make it available to practitioners	Mohr, F., Arnese-Feffin, E., Barolo, M., and Braatz, R. D. (2023). *Smart process analytics for process monitoring* [In preparation].

instrumentation installed on the process equipment. This causes measurements to be available at low frequency, therefore fast, reversible fouling cannot be properly characterized. Furthermore, the process variables used for monitoring are affected by strong variability due to process operation, which hinders the interpretation of fouling state of membranes. The proposed soft sensor delivers high-frequency estimates of the resistances of individual membranes in the multi-module systems based on process variables acquired online. The several advantages of this improved monitoring strategy were discussed, including clear and interpretable dynamics of resistances, decoupling of membrane states (removal of interactions among modules, which affect process variables), and identification of fouling events affecting single modules. A strategy to resolve the effects of reversible and irreversible fouling was proposed as well.

A further step was taken in Chapter 5, which discussed a **systematic analysis of membrane fouling by feature-oriented modeling**. Due to fouling, the membrane process runs in semi-continuous regime, which hinders the application of standard data analytics methods for process understanding. Feature-oriented modeling elegantly solved this issue, while simultaneously incorporating process knowledge in the analysis and simplifying the interpretation of the resulting models. Due to the complexity of the operation, a large number of process settings potentially related to fouling had to be investigated. Therefore, a **systematic procedure for feature screening** aimed at identified the process settings most related to membrane fouling was proposed. The effectiveness of the policies implemented in the plant to counteract reversible fouling and to compensate for the effect of the biomass concentration in the feed was proven. The analysis uncovered a strong interaction between reversible and irreversible fouling, which offered guidelines to improve the maintenance schedule of membranes.

Regarding the advancement of data-driven modeling, an **improved formulation of the algebraic inversion of latent-variable models** was proposed in Chapter 6. Latent-variable model inversion can aid **product design** by identifying the process conditions to manufacture a product with an assigned target quality. However, the most common method currently available requires the variables describing the product quality to be independent, prescribing the removal of correlated variables before model development. No target can be set for the variables not considered in the product design exercise, which could therefore not comply with the acceptable quality specifications. The proposed framework deals with correlation among quality variables by design, addressing the numerical errors arising in the inversion phase by an optimal regularization (in terms of information loss). The advantages of the method were demonstrated on two simulated case studies of fermentation processes. Particular attention was devoted to the estimation of the uncertainties of the inversion solution and of the null space, proving the superiority of the proposed method.

Chapter 7 proposed an **automatic framework for selection and calibration of data-driven methods for fault detection**. Selecting the best model for a given application is a hard task due to the several alternatives available, which feature varying degrees of complexity and rely on

different assumptions. As a result, practitioners typically choose the models they are most accustomed with. The proposed framework automates the selection process to ease the burden on practitioners. Only data from normal operating conditions are required. A **preliminary assessment of the characteristics of the data** is conducted to search for nonlinearity of the correlation among variables (equivalent to non-normality of the distribution of data), dynamics in the data, and availability of variables describing the product quality. Appropriate models (to cope with the found data characteristics) are pre-selected, their hyperparameters tuned, and the best one is identified, all in a **rigorous model selection and discrimination procedure**. Criteria for checking the aforementioned data characteristics were developed and validated with rigorous Monte Carlo studies. The model selection procedure was fine-tuned by design of a **specific figure of merit to measure the performance of fault detection methods** on normal operating conditions data. The effectiveness of the framework was verified on four case studies: a simulated linear, static dataset; the Tennessee Eastman Process simulator; a simulation of a process for continuous filtering and drying of paracetamol; data from an industrial metal etching process for semiconductor manufacturing. In all case studies, the model identified by the proposed framework was the most appropriate one (among those included in the library), showing the best fault detection performance on testing data from faulty conditions (not used for model calibration).

While the results discussed in this Thesis and summarized above represent significant achievements for the digitalization of biorefineries, several areas for future investigation can be identified and are briefly discussed hereby.

- The guidelines for product quality recovery developed in Chapter 3 offer precious indications to steer the bioconversion process as to increase the quality of the product. Experimental validation of such results could not be performed in the study, but it is worth exploring.
- The work on the upstream process presented in Chapter 3 uncovered strong interactions among process variables, which have dramatic effects of the final product quality. The development of digital support systems, such as process monitoring systems or soft sensors for end-of-batch quality prediction, will be beneficial for prompt fault detection and to offer guidelines to operators to take action immediately after a fault occurred.
- The studies carried out in Chapter 4 and Chapter 5 offered valuable insights on the nature and causes of fouling in the ultrafiltration process. The operation of such process can be significantly enhanced by implementing a predictive maintenance system. To this end, a dynamic model of membrane resistances should be developed to predict the future fouling state of the membranes.
- Chapter 4 and Chapter 5 focused on the ultrafiltration operation in the downstream chain, suffering from membrane fouling. The ion-exchange chromatography (Figure 1.3) suffers from similar issues in the form of resin exhaustion. A data-driven analysis

for process understanding and improvement could significantly aid the conduction of this exceptionally complex process, featuring over sixty ion-exchange columns operated in cycled mode.

- All the operations considered in this Thesis, regarding both the upstream and downstream processes, have been found to show clear dynamic evolutions due to many factors, for example sensor drifts, seasonal effects of temperature, membrane fouling, or resin exhaustion. Adaptive data-driven methods, such as automatic model update or re-calibration, to deal with the everchanging nature of the process can provide remarkable benefits in similar scenarios and will be explored in future studies.
- The framework proposed in Chapter 7 relies only on data from normal operating conditions, implementing the so-called rigorous approach to model selection. However, the use of data from faulty conditions for model selection (the so-called compliant approach) could be beneficial in some cases, especially when high-complexity models are chosen. The study of such an approach, including the development of data fusion rules if more than one faulty dataset is available, is matter of future research.

Appendix A

Complete results of Monte Carlo studies for assessment of the dataset properties

The complete results of the Monte Carlo studies discussed in Section 7.4 are reported in this Appendix. Title of each figure is coded as "*⟨criterion⟩* on *⟨case⟩*". Criteria are identified as follows.

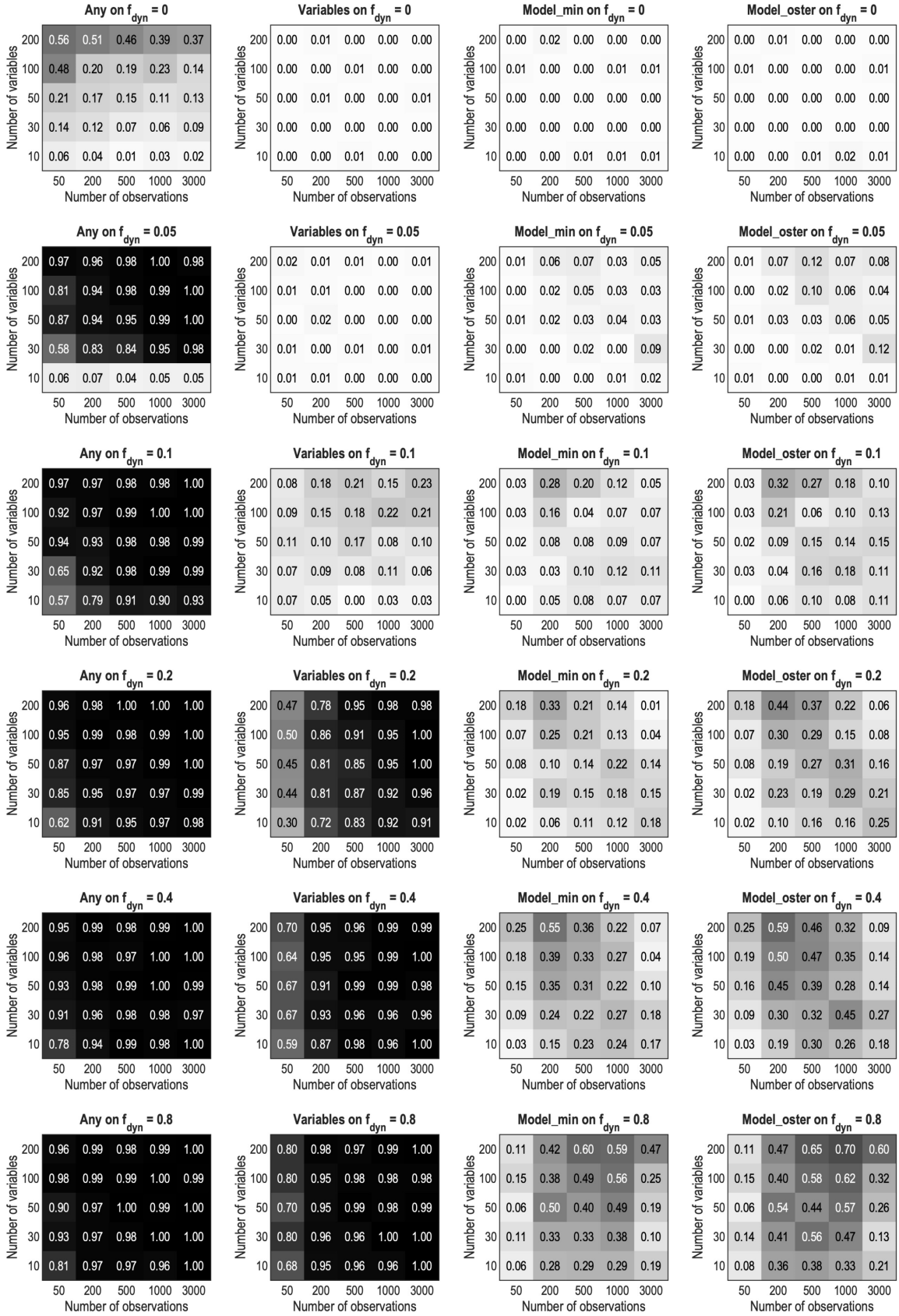
- Combined test: combined test for multivariate normality (see Section 7.4.1).
- Royston: the Royston test for multivariate normality.
- Henze-Zirkler: the Henze-Zirkler test for multivariate normality.
- Mardia combined: the Mardia combined test for multivariate normality.
- Mardia skewness: the Mardia skewness test for multivariate normality.
- Mardia kurtosis: the Mardia Kurtosis test for multivariate normality.
- Non-normality: combined non-normality test applied for nonlinearity detection.
- Any: "any" criterion for nonlinearity detection (see Section 7.4.2) or for dynamics detection (see Section 7.4.3).
- Variables: "variables" criterion for nonlinearity detection (see Section 7.4.2) or for dynamics detection (see Section 7.4.3).
- Couples: "couples" criterion for nonlinearity detection (see Section 7.4.2).
- Model_min: "model_min" criterion for dynamics detection (see Section 7.4.3).
- Model_oster: "model_oster" criterion for dynamics detection (see Section 7.4.3).

Cases are identified as follows.

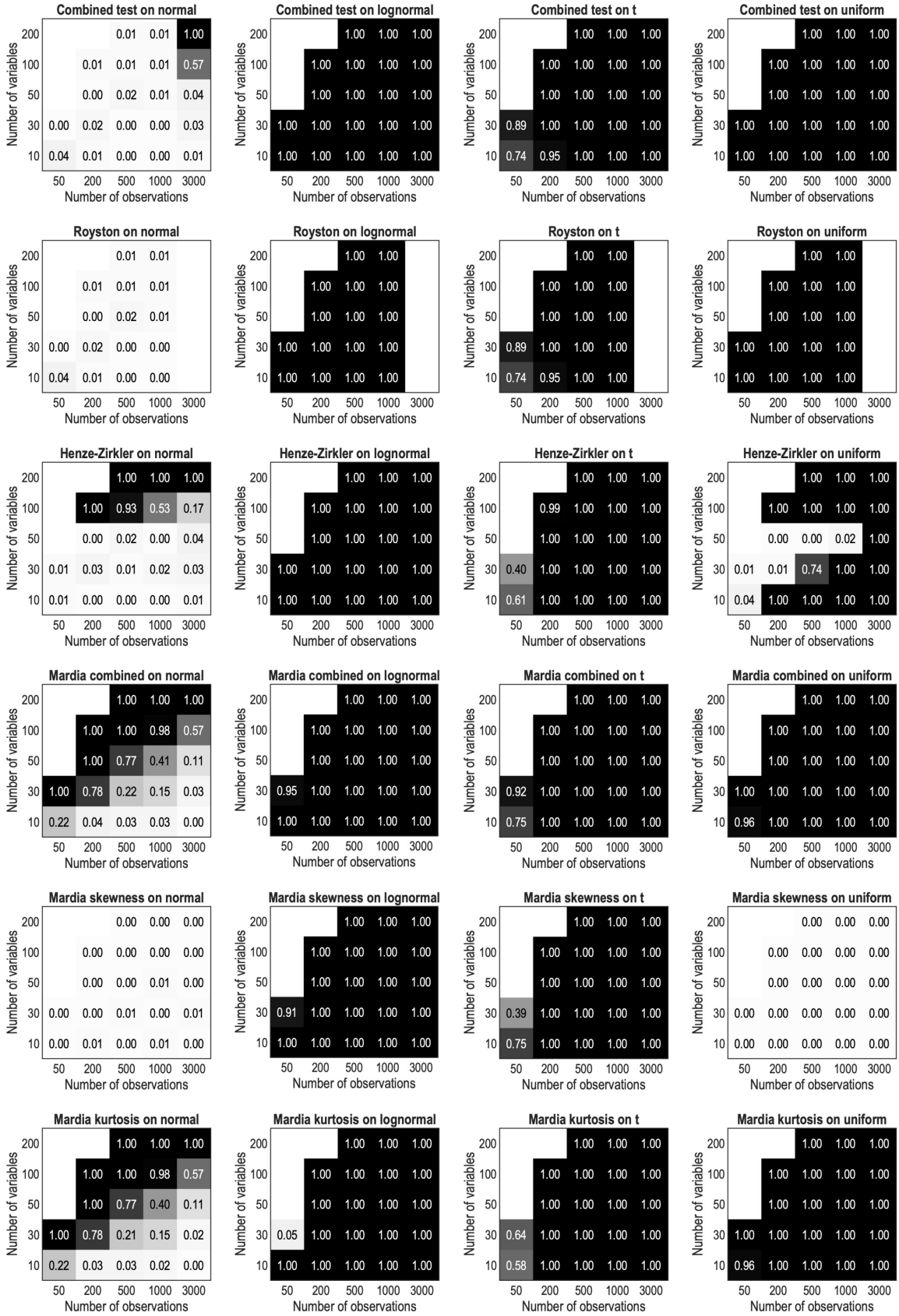
- normal: the dataset is generated from a multivariate normal distribution.
- lognormal: the dataset is generated from a multivariate lognormal distribution.
- t: the dataset is generated from a multivariate t distribution.
- uniform: the dataset is generated from a multivariate uniform distribution.
- f_{nl} = x: $100x\%$ of the variables in the dataset are nonlinearly correlated with the remaining $100(1-x)\%$ linear variables; the dataset is generated as described in Section 7.4.2.
- f_{dyn} = x: $100x\%$ of the variables in the dataset are dynamic, while the remaining $100(1-x)\%$ are static; the dataset is generated as described in Section 7.4.3.

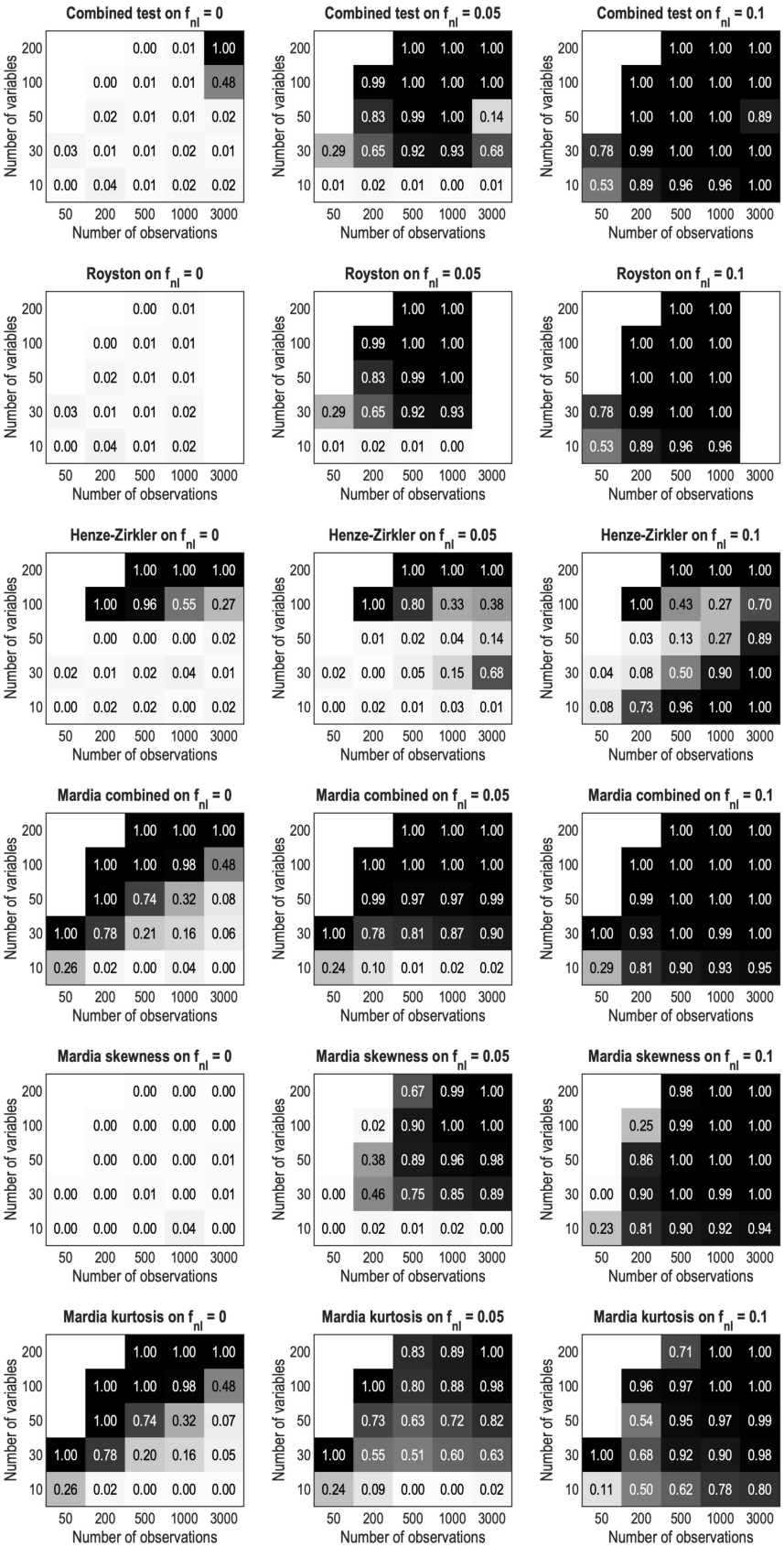
Finally, note that values reported in the heatmaps are the detection rates of non-normality, nonlinearity, and dynamics as defined in Sections 7.4.1, 7.4.2, and 7.4.3, respectively.

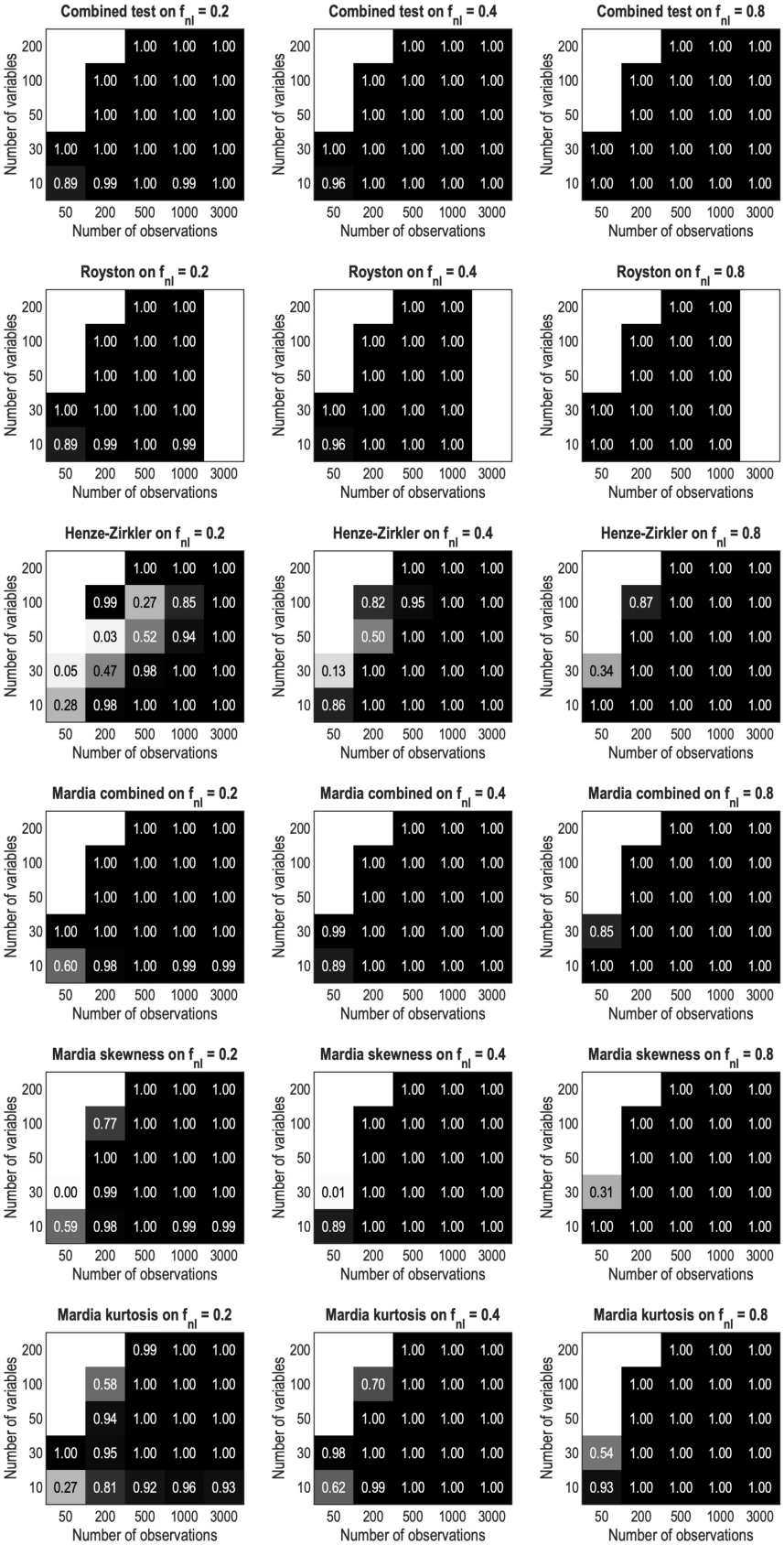
A.1 Results of the Monte Carlo study on dynamics detection



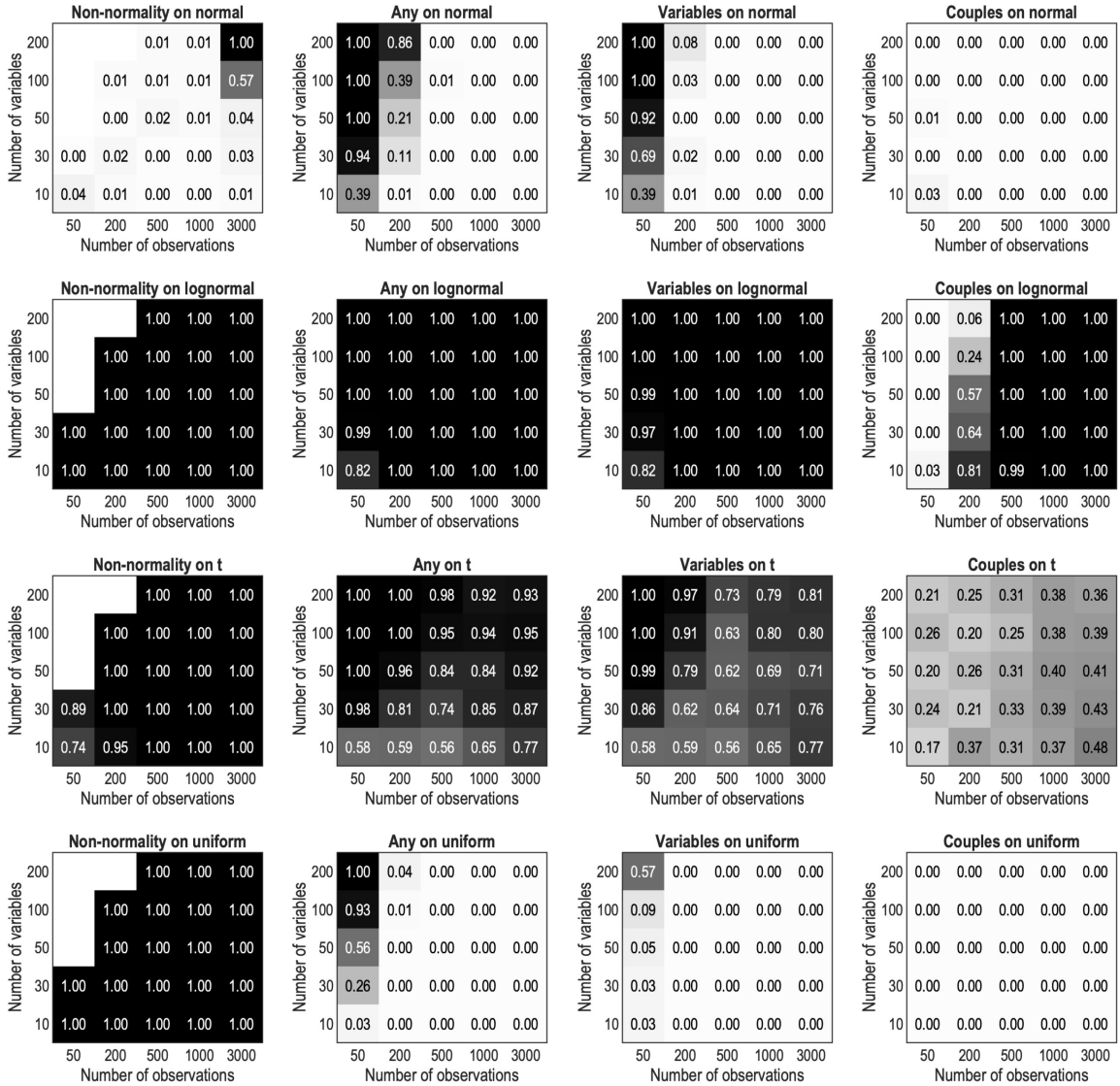
A.2 Results of the Monte Carlo study on non-normality detection

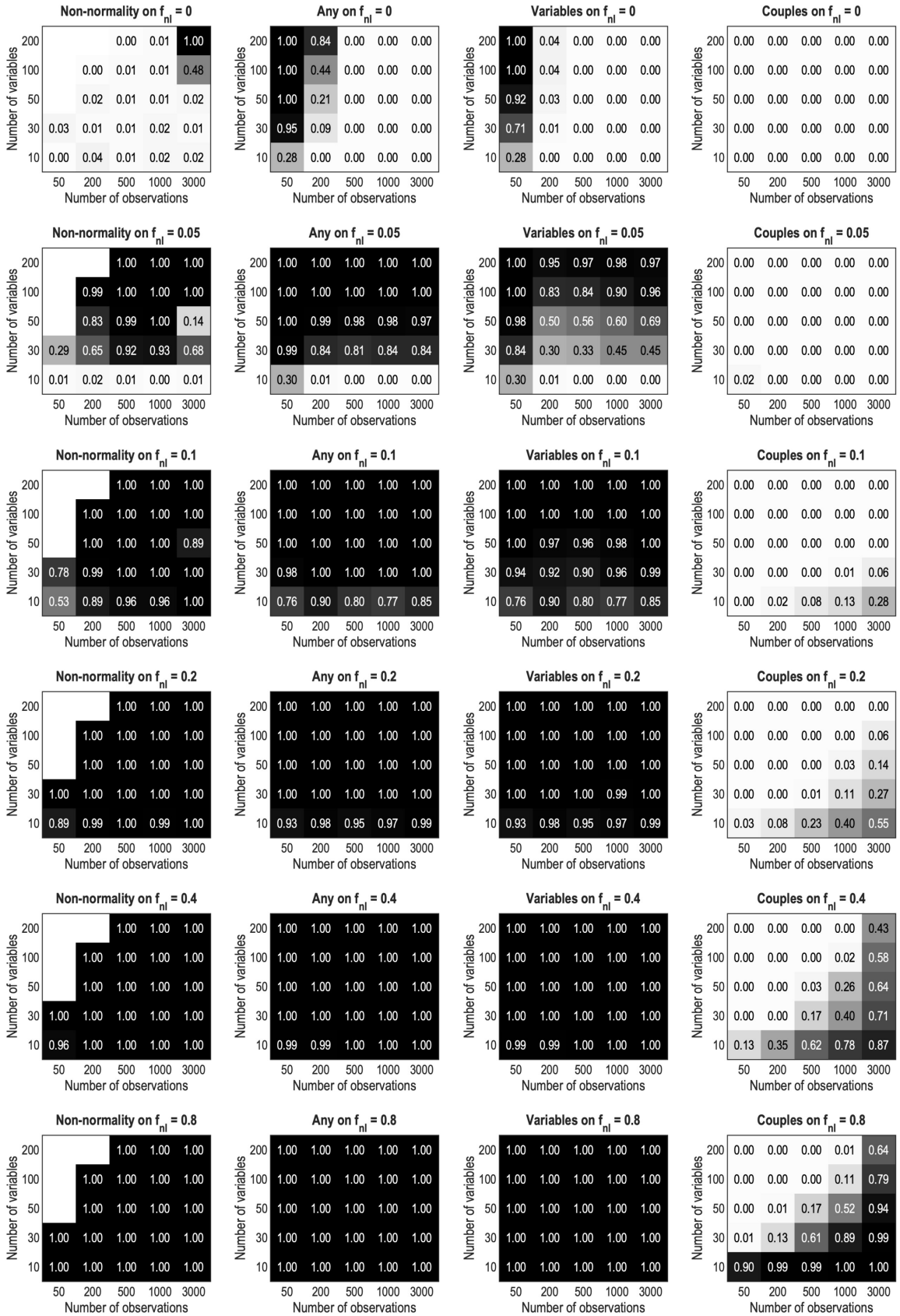






A.3 Results of the Monte Carlo study on nonlinearity detection





References

Abels, C., Carstensen, F., and Wessling, M. (2013). Membrane processes in biorefinery applications. *Journal of Membrane Science*, **444**, 285–317. https://doi.org/10.1016/j.memsci.2013.05.030

Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle. *Proceedings of the 2nd International Symposium on Information Theory*. https://doi.org/10.1007/978-1-4612-1694-0_15

Alam, F., Mobin, S., and Chowdhury, H. (2015). Third Generation Biofuel from Algae. *Procedia Engineering*, **105**, 763–768. https://doi.org/10.1016/j.proeng.2015.05.068

Allen, D. M. (1974). The Relationship Between Variable Selection and Data Agumentation and a Method for Prediction. *Technometrics*, **16**(1), 125–127. https://doi.org/10.1080/00401706.1974.10489157

AlSawaftah, N., Abuwatfa, W., Darwish, N., and Husseini, G. (2021). A Comprehensive Review on Membrane Fouling: Mathematical Modelling, Prediction, Diagnosis, and Mitigation. *Water*, **13**(9), 1327. https://doi.org/10.3390/w13091327

Amerit, B., Ntayi, J. M., Ngoma, M., Bashir, H., Echegu, S., and Nantongo, M. (2023). Commercialization of biofuel products: A systematic literature review. *Renewable Energy Focus*, **44**, 223–236. https://doi.org/10.1016/j.ref.2022.12.008

Andersen, S. W., and Runger, G. C. (2012). Automated feature extraction from profiles with application to a batch fermentation process. *Journal of the Royal Statistical Society. Series C: Applied Statistics*, **61**(2), 327–344. https://doi.org/10.1111/j.1467-9876.2011.01032.x

Andersson, M. (2009). A comparison of nine PLS1 algorithms. *Journal of Chemometrics*, **23**(10), 518–529. https://doi.org/10.1002/cem.1248

Arce, M. M., Ruiz, S., Sanllorente, S., Ortiz, M. C., Sarabia, L. A., and Sánchez, M. S. (2021). A new approach based on inversion of a partial least squares model searching for a preset analytical target profile. Application to the determination of five bisphenols by liquid chromatography with diode array detector. *Analytica Chimica Acta*, **1149**, 338217. https://doi.org/10.1016/j.aca.2021.338217

Arlot, S., and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, **4**, 40–79. https://doi.org/10.1214/09-SS054

Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2022). Digital design of new products: Accounting for output correlation via a novel algebraic formulation of the latent-variable model inversion problem. *Chemometrics and Intelligent Laboratory Systems*, **227**(June), 104610. https://doi.org/10.1016/j.chemolab.2022.104610

Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2023a). Estimation of null-space uncertainty in latent-variable model inversion: The case of correlated quality attributes. *XI Colloquium Chemometricum Mediterraneum – Book of Abstracts*, 144–145.

Arnese-Feffin, E., Facco, P., Bezzo, F., and Barolo, M. (2023b, June 27). *Estimation of null-space uncertainty in latent-variable model inversion: The case of correlated quality attributes* [Poster presentation]. XI Colloquium Chemometricum Mediterraneum – CCM 2023, Padova (IT).

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023c). Troubleshooting high-pressure issues in an industrial biorefinery process by feature-oriented modeling. *Proceedings of the 33rd European Symposium on Computer Aided Process Engineering (ESCAPE33)*, 163–168. https://doi.org/10.1016/B978-0-443-15274-0.50027-5

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023d). *Understanding fouling in an industrial biorefinery membrane separation process by feature-oriented data-driven modeling* [In preparation].

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2024). Hybrid modeling of a biorefinery separation process for performance monitoring. *Chemical Engineering Science*, **283**, 119413. https://doi.org/10.1016/j.ces.2023.119413

Arnese-Feffin, E., Facco, P., Turati, D., Bezzo, F., and Barolo, M. (2023e, June 20). *Troubleshooting high-pressure issues in an industrial biorefinery process by feature-oriented modeling* [Oral presentation]. 33rd European Symposium on Computer-Aided Chemical Engineering – ESCAPE 2023, Athens (GR).

Attard, T. M., Clark, J. H., and McElroy, C. R. (2020). Recent developments in key biorefinery areas. *Current Opinion in Green and Sustainable Chemistry*, **21**, 64–74. https://doi.org/10.1016/j.cogsc.2019.12.002

Azamfar, M., Li, X., and Lee, J. (2020). Deep Learning-Based Domain Adaptation Method for Fault Diagnosis in Semiconductor Manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, **33**(3), 445–453. https://doi.org/10.1109/TSM.2020.2995548

Azcarate, S. M., Ríos-Reina, R., Amigo, J. M., and Goicoechea, H. C. (2021). Data handling in data fusion: Methodologies and applications. *TrAC Trends in Analytical Chemistry*, **143**, 116355. https://doi.org/10.1016/j.trac.2021.116355

Baffi, G., Martin, E. B., and Morris, A. J. (2000). Non-linear dynamic projection to latent structures modelling. *Chemometrics and Intelligent Laboratory Systems*, **52**(1), 5–22. https://doi.org/10.1016/S0169-7439(00)00083-6

Bagheri, M., Akbari, A., and Mirbagheri, S. A. (2019). Advanced control of membrane fouling in filtration systems using artificial intelligence and machine learning techniques: A critical review. *Process Safety and Environmental Protection*, **123**, 229–252. https://doi.org/10.1016/j.psep.2019.01.013

Bähner, F. D., Prado-Rubio, O. A., and Huusom, J. K. (2021). Challenges in Optimization and Control of Biobased Process Systems: An Industrial-Academic Perspective. *Industrial & Engineering Chemistry Research*, **60**(42), 14985–15003. https://doi.org/10.1021/acs.iecr.1c01792

Bähner, F. D., Santacoloma, P. A., and Huusom, J. K. (2019). Optimal operation of parallel dead-end filters in a continuous bio-based process. *Food and Bioproducts Processing*, **114**, 263–275. https://doi.org/10.1016/j.fbp.2019.02.001

Baidya, P. K., Sarkar, U., Villa, R., and Sadhukhan, S. (2019). Liquid-phase hydrogenation of bio-refined succinic acid to 1,4-butanediol using bimetallic catalysts. *BMC Chemical Engineering*, **1**(1), 10. https://doi.org/10.1186/s42480-019-0010-z

Baker, R. W. (2004). *Membrane Technology and Applications* (2nd ed.). Wiley.

Bakshi, B. R. (1998). Multiscale PCA with application to multivariate statistical process monitoring. *AIChE Journal*, **44**(7), 1596–1610. https://doi.org/10.1002/aic.690440712

Bakshi, B. R., and Stephanopoulos, G. (1996). Compression of chemical process data by functional approximation and feature extraction. *AIChE Journal*, **42**(2), 477–492. https://doi.org/10.1002/aic.690420217

Baldoni, E., Reumermann, P., Parisi, C., Platt, R., González-Hermoso, H., Vikla, K., Vos, J., and M'barek, R. (2021a). *Chemical and material biorefineries in the EU*. http://data.europa.eu/89h/24e98d11-ef06-4233-8f69-1e123938e891

Baldoni, E., Reumermann, P., Parisi, C., Platt, R., González-Hermoso, H., Vikla, K., Vos, J., and M'barek, R. (2021b). *Chemical and material biorefineries outside the EU*. http://data.europa.eu/89h/bd890922-fc66-41ca-b032-e02754b34a22

Ballabio, D., and Consonni, V. (2013). Classification tools in chemistry. Part 1: Linear models. PLS-DA. *Analytical Methods*, **5**(16), 3790–3798. https://doi.org/10.1039/c3ay40582f

Bano, G., Facco, P., Bezzo, F., and Barolo, M. (2018a). Probabilistic Design space determination in pharmaceutical product development: A Bayesian/latent variable approach. *AIChE Journal*, **64**(7), 2438–2449. https://doi.org/10.1002/aic.16133

Bano, G., Facco, P., Meneghetti, N., Bezzo, F., and Barolo, M. (2017). Uncertainty back-propagation in PLS model inversion for design space determination in pharmaceutical product development. *Computers and Chemical Engineering*, **101**, 110–124. https://doi.org/10.1016/j.compchemeng.2017.02.038

Bano, G., Wang, Z., Facco, P., Bezzo, F., Barolo, M., and Ierapetritou, M. (2018b). A novel and systematic approach to identify the design space of pharmaceutical processes. *Computers and Chemical Engineering*, **115**, 309–322. https://doi.org/10.1016/j.compchemeng.2018.04.021

Barker, M., and Rayens, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics*, **17**(3), 166–173. https://doi.org/10.1002/cem.785

Barragán-Ocaña, A., Merritt, H., Sánchez-Estrada, O. E., Méndez-Becerril, J. L., and del Pilar

Longar-Blanco, M. (2023). Biorefinery and sustainability for the production of biofuels and value-added products: A trends analysis based on network and patent analysis. *PLOS ONE*, **18**(1), e0279659. https://doi.org/10.1371/journal.pone.0279659

Bartlett, M. S. (1946). On the Theoretical Specification and Sampling Properties of Autocorrelated Time-Series. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, **8**, 27–41. https://doi.org/10.2307/2983611

Barton, N. R., Burgard, A. P., Burk, M. J., Crater, J. S., Osterhout, R. E., Pharkya, P., Steer, B. A., Sun, J., Trawick, J. D., Van Dien, S. J., Yang, T. H., and Yim, H. (2015). An integrated biotechnology platform for developing sustainable chemical processes. *Journal of Industrial Microbiology and Biotechnology*, **42**(3), 349–360. https://doi.org/10.1007/s10295-014-1541-1

Berber, R., and Akcay, L. (2005). Monitoring and fault diagnosis by multivariate statistical methods in chemical processes. *AIChE Annual Meeting, Conference Proceedings*, 6855.

Bergmeir, C., and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, **191**, 192–213. https://doi.org/10.1016/j.ins.2011.12.028

Birol, G., Ündey, C., and Çinar, A. (2002). A modular simulation package for fed-batch fermentation: Penicillin production. *Computers and Chemical Engineering*, **26**(11), 1553–1565. https://doi.org/10.1016/S0098-1354(02)00127-8

Bishop, C. M. (1995). *Neural Network for Pattern Recognition* (4th ed.). Clarendon Press.

Blasius, J., and Gower, J. C. (2005). Multivariate Prediction with Nonlinear Principal Components Analysis: Application. *Quality & Quantity*, **39**(4), 373–390. https://doi.org/10.1007/s11135-005-3006-0

Bodor, Z., Tompos, L., Nechifor, A. C., and Bodor, K. (2019). In silico Analysis of 1,4-butanediol Heterologous Pathway Impact on Escherichia coli Metabolism. *Revista de Chimie*, **70**(10), 3448–3455. https://doi.org/10.37358/rc.19.10.7574

Bolton, G., LaCasse, D., and Kuriyel, R. (2006). Combined models of membrane fouling: Development and application to microfiltration and ultrafiltration of biological fluids. *Journal of Membrane Science*, **277**(1–2), 75–84. https://doi.org/10.1016/j.memsci.2004.12.053

Box, G. E. P. (1954). Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, II. Effects of Inequality of Variance and of Correlation Between Errors in the Two-Way Classification. *The Annals of Mathematical Statistics*, **25**(3), 484–498. https://doi.org/10.1214/aoms/1177728717

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2016). *Time Series Analysis*. Wiley.

Breiman, L., and Friedman, J. H. (1985). Estimating optimal transformations for multiple

regression and correlation. *Journal of the American Statistical Association*, **80**(391), 580–598. https://doi.org/10.1080/01621459.1985.10478157

Breiman, L., and Spector, P. (1992). Submodel Selection and Evaluation in Regression. The X-Random Case. *International Statistical Review*, **60**(3), 291. https://doi.org/10.2307/1403680

Brereton, R. G. (2011). One-class classifiers. *Journal of Chemometrics*, **25**(5), 225–246. https://doi.org/10.1002/cem.1397

Bro, R., Kjeldahl, K., Smilde, A. K., and Kiers, H. A. L. (2008). Cross-validation of component models: A critical look at current methods. *Analytical and Bioanalytical Chemistry*, **390**(5), 1241–1251. https://doi.org/10.1007/s00216-007-1790-1

Bro, R., and Smilde, A. K. (2003). Centering and scaling in component analysis. *Journal of Chemometrics*, **17**(1), 16–33. https://doi.org/10.1002/cem.773

Brockwell, P. J., and Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. Springer.

Burg, J. M., Cooper, C. B., Ye, Z., Reed, B. R., Moreb, E. A., and Lynch, M. D. (2016). Large-scale bioprocess competitiveness: The potential of dynamic metabolic control in two-stage fermentations. *Current Opinion in Chemical Engineering*, **14**, 121–136. https://doi.org/10.1016/j.coche.2016.09.008

Burgard, A., Burk, M. J., Osterhout, R., Van Dien, S., and Yim, H. (2016). Development of a commercial scale process for production of 1,4-butanediol from sugar. *Current Opinion in Biotechnology*, **42**, 118–125. https://doi.org/10.1016/j.copbio.2016.04.016

Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, **76**(3), 503–514. https://doi.org/10.1093/biomet/76.3.503

Burnham, A. J., MacGregor, J. F., and Viveros, R. (1999). Latent variable multivariate regression modeling. *Chemometrics and Intelligent Laboratory Systems*, **48**(2), 167–180. https://doi.org/10.1016/S0169-7439(99)00018-0

Burnham, A. J., MacGregor, J. F., and Viveros, R. (2001). Interpretation of regression coefficients under a latent variable regression model. *Journal of Chemometrics*, **15**(4), 265–284. https://doi.org/10.1002/cem.680

Burnham, A. J., Viveros, R., and Macgregor, J. F. (1996). Frameworks for latent variable multivariate regression. *Journal of Chemometrics*, **10**(1), 31–45. https://doi.org/10.1002/(SICI)1099-128X(199601)10:1<31::AID-CEM398>3.0.CO;2-1

Burnham, K. P., and Anderson, D. R. (2002). *Model selection and multimodel inference* (2nd ed.). Springer.

Camacho, J., and Ferrer, A. (2012). Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Theoretical aspects. *Journal of Chemometrics*, **26**(7), 361–373.

https://doi.org/10.1002/cem.2440

Camacho, J., and Ferrer, A. (2014). Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Practical aspects. *Chemometrics and Intelligent Laboratory Systems*, **131**, 37–50. https://doi.org/10.1016/j.chemolab.2013.12.003

Camacho, J., Pérez-Villegas, A., Garciá-Teodoro, P., and MacIá-Fernández, G. (2016). PCA-based multivariate statistical network monitoring for anomaly detection. *Computers and Security*, **59**, 118–137. https://doi.org/10.1016/j.cose.2016.02.008

Camacho, J., and Picó, J. (2006a). Multi-phase principal component analysis for batch processes modelling. *Chemometrics and Intelligent Laboratory Systems*, **81**(2), 127–136. https://doi.org/10.1016/j.chemolab.2005.11.003

Camacho, J., and Picó, J. (2006b). Online monitoring of batch processes using multi-phase principal component analysis. *Journal of Process Control*, **16**(10), 1021–1035. https://doi.org/10.1016/j.jprocont.2006.07.005

Camacho, J., Picó, J., and Ferrer, A. (2008a). Bilinear modelling of batch processes. Part I: Theoretical discussion. *Journal of Chemometrics*, **22**(5), 299–308. https://doi.org/10.1002/cem.1113

Camacho, J., Picó, J., and Ferrer, A. (2008b). Bilinear modelling of batch processes. Part II: A comparison of PLS soft-sensors. *Journal of Chemometrics*, **22**(10), 533–547. https://doi.org/10.1002/cem.1179

Camacho, J., Picó, J., and Ferrer, A. (2009). The best approaches in the on-line monitoring of batch processes based on PCA: Does the modelling structure matter? *Analytica Chimica Acta*, **642**(1–2), 59–68. https://doi.org/10.1016/j.aca.2009.02.001

Camacho, J., Picó, J., and Ferrer, A. (2010). Data understanding with PCA: Structural and Variance Information plots. *Chemometrics and Intelligent Laboratory Systems*, **100**(1), 48–56. https://doi.org/10.1016/j.chemolab.2009.10.005

Candy, J. V., Bullock, T. E., and Warren, M. E. (1979). Invariant System Description of the Stochastic Realization. *Automatica*, **15**, 493–495. https://doi.org/10.1016/0005-1098(79)90026-8

Carinhas, N., Bernal, V., Teixeira, A. P., Carrondo, M. J., Alves, P. M., and Oliveira, R. (2011). Hybrid metabolic flux analysis: Combining stoichiometric and statistical constraints to model the formation of complex recombinant products. *BMC Systems Biology*, **5**(1), 34. https://doi.org/10.1186/1752-0509-5-34

Carstensen, F., Apel, A., and Wessling, M. (2012). In situ product recovery: Submerged membranes vs. External loop membranes. *Journal of Membrane Science*, **394–395**, 1–36. https://doi.org/10.1016/j.memsci.2011.11.029

Chan, L. L. T., Chou, C.-P., and Chen, J. (2017). Hybrid model based expected improvement control for cyclical operation of membrane microfiltration processes. *Chemical Engineering Science*, **166**, 77–90. https://doi.org/10.1016/j.ces.2017.02.048

Chapra, S. P., and Canale, R. P. (2015). *Numerical Methods for Engineers* (7th ed.). McGraw Hill.

Chatfiled, C., and Xing, H. (2019). *The analysis of time series*. CRC Press.

Chen, F., Peldszus, S., Peiris, R. H., Ruhl, A. S., Mehrez, R., Jekel, M., Legge, R. L., and Huck, P. M. (2014). Pilot-scale investigation of drinking water ultrafiltration membrane fouling rates using advanced data analysis techniques. *Water Research*, **48**(1), 508–518. https://doi.org/10.1016/j.watres.2013.10.007

Chen, J., and Liu, K. C. (2002). On-line batch process monitoring using dynamic PCA and dynamic PLS models. *Chemical Engineering Science*, **57**(1), 63–75. https://doi.org/10.1016/S0009-2509(01)00366-9

Chen, T., and Chen, H. (1995). Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems. *IEEE Transactions on Neural Networks*, **6**(4), 911–917. https://doi.org/10.1109/72.392253

Chen, T., and Zhang, J. (2010). On-line multivariate statistical monitoring of batch processes using Gaussian mixture model. *Computers and Chemical Engineering*, **34**(4), 500–507. https://doi.org/10.1016/j.compchemeng.2009.08.007

Cheng, J., Li, J., and Zheng, L. (2021). Achievements and Perspectives in 1,4-Butanediol Production from Engineered Microorganisms. *Journal of Agricultural and Food Chemistry*, **69**(36), 10480–10485. https://doi.org/10.1021/acs.jafc.1c03769

Cherry, G. A., and Qin, S. J. (2007). Monitoring Non-normal Data with Principal Component Analysis and Adaptive Density Estimation. *Proceedings of the 46th IEEE Conference on Decision and Control*, 352–359.

Cherubini, F. (2010). The biorefinery concept: Using biomass instead of oil for producing energy and chemicals. *Energy Conversion and Management*, **51**(7), 1412–1421. https://doi.org/10.1016/j.enconman.2010.01.015

Chew, C. M., Aroua, M. K., and Hussain, M. A. (2017). A practical hybrid modelling approach for the prediction of potential fouling parameters in ultrafiltration membrane water treatment plant. *Journal of Industrial and Engineering Chemistry*, **45**, 145–155. https://doi.org/10.1016/j.jiec.2016.09.017

Chiang, L. H., Leardi, R., Pell, R. J., and Seasholtz, M. B. (2006). Industrial experiences with multivariate statistical analysis of batch process data. *Chemometrics and Intelligent Laboratory Systems*, **81**(2), 109–119. https://doi.org/10.1016/j.chemolab.2005.10.006

Chiang, L. H., Russell, E. L., and Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems* (1st ed.). Springer. https://doi.org/10.1088/0957-0233/12/10/706

Cho, J. H., Lee, J. M., Choi, S. W., Lee, D., and Lee, I. B. (2005). Fault identification for process monitoring using kernel principal component analysis. *Chemical Engineering Science*, **60**(1), 279–288. https://doi.org/10.1016/j.ces.2004.08.007

Choi, S., Song, H., Lim, S. W., Kim, T. Y., Ahn, J. H., Lee, J. W., Lee, M.-H., and Lee, S. Y. (2016). Highly selective production of succinic acid by metabolically engineered *Mannheimia succiniciproducens* and its efficient purification: Nearly Homo-Succinic Acid Fermentation With High Productivity. *Biotechnology and Bioengineering*, **113**(10), 2168–2177. https://doi.org/10.1002/bit.25988

Choi, S. W., Lee, C., Lee, J. M., Park, J. H., and Lee, I. B. (2005). Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, **75**(1), 55–67. https://doi.org/10.1016/j.chemolab.2004.05.001

Choi, S. W., and Lee, I. B. (2004). Nonlinear dynamic process monitoring based on dynamic kernel PCA. *Chemical Engineering Science*, **59**(24), 5897–5908. https://doi.org/10.1016/j.ces.2004.07.019

Chu, F., Wang, J., Zhao, X., Zhang, S., Chen, T., Jia, R., and Xiong, G. (2021). Transfer learning for nonlinear batch process operation optimization. *Journal of Process Control*, **101**, 11–23. https://doi.org/10.1016/j.jprocont.2021.03.002

Cremers, D., Kohlberger, T., and Schnörr, C. (2003). Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, **36**(9), 1929–1943. https://doi.org/10.1016/S0031-3203(03)00056-6

Cuellar, M. C., and Straathof, A. J. (2018). Improving Fermentation by Product Removal. In *Intensification of Biobased Processes* (1st ed., pp. 86–108). Royal Society of Chemistry.

Cuellar, M. C., and Straathof, A. J. (2020). Downstream of the bioreactor: Advancements in recovering fuels and commodity chemicals. *Current Opinion in Biotechnology*, **62**, 189–195. https://doi.org/10.1016/j.copbio.2019.11.012

Culaba, A. B., Mayol, A. P., San Juan, J. L. G., Vinoya, C. L., Concepcion, R. S., Bandala, A. A., Vicerra, R. R. P., Ubando, A. T., Chen, W.-H., and Chang, J.-S. (2022). Smart sustainable biorefineries for lignocellulosic biomass. *Bioresource Technology*, **344**, 126215. https://doi.org/10.1016/j.biortech.2021.126215

Dal-Pastro, F., Facco, P., Zamprogna, E., Bezzo, F., and Barolo, M. (2017). Model-based approach to the design and scale-up of wheat milling operations—Proof of concept. *Food and Bioproducts Processing*, **106**, 127–136. https://doi.org/10.1016/j.fbp.2017.09.005

Darcy, H. (1856). *Les Fontaines publiques de la ville de Dijon*. V. Dalmont.

Das, A., Maiti, J., and Banerjee, R. N. (2012). Process monitoring and fault detection strategies: A review. *International Journal of Quality & Reliability Management*, **29**(7), 720–752. https://doi.org/10.1108/02656711211258508

Davies, P. I., and Higham, N. J. (2000). Numerically stable generation of correlation matrices and their factors. *BIT Numerical Mathematics*, **40**(4), 640–651. https://doi.org/10.1023/A:1022384216930

De Bari, I., Giuliano, A., Petrone, M. T., Stoppiello, G., Fatta, V., Giardi, C., Razza, F., and Novelli, A. (2020). From Cardoon Lignocellulosic Biomass to Bio-1,4 Butanediol: An Integrated Biorefinery Model. *Processes*, **8**(12), 1585. https://doi.org/10.3390/pr8121585

de Jong, S. (1993). SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, **18**, 251–263. https://doi.org/10.1016/0169-7439(93)85002-X

Delbecq, F., Wang, Y., Muralidhara, A., El Ouardi, K., Marlair, G., and Len, C. (2018). Hydrolysis of Hemicellulose and Derivatives—A Review of Recent Advances in the Production of Furfural. *Frontiers in Chemistry*, **6**, 146. https://doi.org/10.3389/fchem.2018.00146

Delgenes, J. (1996). Comparative study of separated fermentations and cofermentation processes to produce ethanol from hardwood derived hydrolysates. *Biomass and Bioenergy*, **11**(4), 353–360. https://doi.org/10.1016/0961-9534(96)00019-0

Deng, B. C., Yun, Y. H., Liang, Y. Z., Cao, D. S., Xu, Q. S., Yi, L. Z., and Huang, X. (2015). A new strategy to prevent over-fitting in partial least squares models based on model population analysis. *Analytica Chimica Acta*, **880**, 32–41. https://doi.org/10.1016/j.aca.2015.04.045

Destro, F., Facco, P., García Muñoz, S., Bezzo, F., and Barolo, M. (2020). A hybrid framework for process monitoring: Enhancing data-driven methodologies with state and parameter estimation. *Journal of Process Control*, **92**, 333–351. https://doi.org/10.1016/j.jprocont.2020.06.002

Destro, F., Hur, I., Wang, V., Abdi, M., Feng, X., Wood, E., Coleman, S., Firth, P., Barton, A., Barolo, M., and Nagy, Z. K. (2021). Mathematical modeling and digital design of an intensified filtration-washing-drying unit for pharmaceutical continuous manufacturing. *Chemical Engineering Science*, **244**, 116803. https://doi.org/10.1016/j.ces.2021.116803

Destro, F., Nagy, Z. K., and Barolo, M. (2022a). A benchmark simulator for quality-by-design and quality-by-control studies in continuous pharmaceutical manufacturing – Intensified filtration-drying of crystallization slurries. *Computers and Chemical Engineering*, **163**, 107809. https://doi.org/10.1016/j.compchemeng.2022.107809

Destro, F., Nagy, Z. K., and Barolo, M. (2022b). *Continuous Carousel Simulator* (0.9.0) [Computer software]. https://github.com/CryPTSyS/ContCarSim

Diaz-Bejarano, E., Coletti, F., and Macchietto, S. (2020). A Model-Based Method for Visualization, Monitoring, and Diagnosis of Fouling in Heat Exchangers. *Industrial & Engineering Chemistry Research*, **59**(10), 4602–4619. https://doi.org/10.1021/acs.iecr.9b05490

Dologlu, P., and Sildir, H. (2022). Data driven identification of industrial reverse osmosis membrane process. *Computers & Chemical Engineering*, **161**, 107782.

https://doi.org/10.1016/j.compchemeng.2022.107782

Dong, D., and Mcavoy, T. J. (1996). Nonlinear principal component analysis—Based on principal curves and neural networks. *Computers and Chemical Engineering*, **20**(1), 65–78. https://doi.org/10.1016/0098-1354(95)00003-K

Dong, Y., and Joe Qin, S. (2020). New Dynamic Predictive Monitoring Schemes Based on Dynamic Latent Variable Models. *Industrial and Engineering Chemistry Research*, **59**(6), 2353–2365. https://doi.org/10.1021/acs.iecr.9b04741

Dong, Y., and Qin, S. J. (2015). Dynamic-inner partial least squares for dynamic data modeling. *IFAC-PapersOnLine*, **28**(8), 117–122. https://doi.org/10.1016/j.ifacol.2015.08.167

Dong, Y., and Qin, S. J. (2018a). A novel dynamic PCA algorithm for dynamic data modeling and process monitoring. *Journal of Process Control*, **67**, 1–11. https://doi.org/10.1016/j.jprocont.2017.05.002

Dong, Y., and Qin, S. J. (2018b). Regression on dynamic PLS structures for supervised learning of dynamic data. *Journal of Process Control*, **68**, 64–72. https://doi.org/10.1016/j.jprocont.2018.04.006

Dong, Y., and Qin, S. J. (2021). Adaptive dynamic predictive monitoring scheme based on DLV models. *IFAC-PapersOnLine*, **54**(7), 91–96. https://doi.org/10.1016/j.ifacol.2021.08.340

Downs, J. J., and Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering*, **17**(3), 245–255. https://doi.org/10.1016/0098-1354(93)80018-I

Eastment, H. T., and Krzanowski, W. J. (1982). Cross-validatory choice of the number of components from a principal component analysis. *Technometrics*, **24**(1), 73–77. https://doi.org/10.1080/00401706.1982.10487712

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, **7**(1), 1–26. https://doi.org/10.1214/aos/1176344552

Efron, B. (1987). Better Bootstrap Confidence Intervals. *Journal of the American Statistical Association*, **82**, 171–185.

Efron, B., and Tibshirani, R. (1993). *An Introduction to the Bootstrap* (1st ed.). Springer.

Eigenvector Research, Inc. (1999). *Semiconductor metal etch* [dataset]. https://eigenvector.com/resources/data-sets/#metal-sec

Ennaceri, H., Fischer, K., Schulze, A., and Moheimani, N. R. (2022). Membrane fouling control for sustainable microalgal biodiesel production: A review. *Renewable and Sustainable Energy Reviews*, **161**, 112335. https://doi.org/10.1016/j.rser.2022.112335

Ergon, R. (2004). Informative PLS score-loading plots for process understanding and monitoring. *Journal of Process Control*, **14**(8), 889–897. https://doi.org/10.1016/j.jprocont.2004.02.004

European Commission, Directorate General for Research and Innovation, E4tech, WUR, BTG,

FNR, ICONS, Platt, R., Bauen, A., Reumermann, P., Geier, C., Van Ree, R., Vural Gursel, I., Garcia, L., Behrens, M., von Bothmer, P., Howes, J., Panchaksharam, Y., Vikla, K., Sartorius, V., and Annevelink, B. (2021). *EU biorefinery outlook to 2030: Studies on support to research and innovation policy in the area of bio based products and services.* Publications Office of the European Union. https://data.europa.eu/doi/10.2777/103465

European Commission, Joint Research Centre, Baldoni, E., Reumermann, P., Parisi, C., Platt, R., González-Hermoso, H., Vikla, K., Vos, J., and M'barek, R. (2021). *Chemical and material driven biorefineries in the EU and beyond*. Publications Office. https://data.europa.eu/doi/10.2760/8932

European Commission, Joint Research Centre, and Parisi, C. (2018). *Biorefineries distribution in the EU*. Publications Office of the European Union. https://data.europa.eu/doi/10.2760/126478

Faber, K., and Kowalski, B. R. (1997). Propagation of measurement errors for the validation of predictions obtained by principal component regression and partial least squares. *Journal of Chemometrics*, **11**(3), 181–238. https://doi.org/10.1002/(SICI)1099-128X(199705)11:3<181::AID-CEM459>3.0.CO;2-7

Faber, N. M., and Bro, R. (2002). Standard error of prediction for multiway PLS - 1. Background and a simulation study. *Chemometrics and Intelligent Laboratory Systems*, **61**(1–2), 133–149. https://doi.org/10.1016/S0169-7439(01)00204-0

Facco, P., Dal Pastro, F., Meneghetti, N., Bezzo, F., and Barolo, M. (2015). Bracketing the Design Space within the Knowledge Space in Pharmaceutical Product Development. *Industrial and Engineering Chemistry Research*, **54**(18), 5128–5138. https://doi.org/10.1021/acs.iecr.5b00863

Facco, P., Largoni, M., Tomba, E., Bezzo, F., and Barolo, M. (2014). Transfer of process monitoring models between plants: Batch systems. *Chemical Engineering Research and Design*, **92**(2), 273–284. https://doi.org/10.1016/j.cherd.2013.07.010

Facco, P., Tomba, E., Bezzo, F., García Muñoz, S., and Barolo, M. (2012). Transfer of process monitoring models between different plants using latent variable techniques. *Industrial and Engineering Chemistry Research*, **51**(21), 7327–7339. https://doi.org/10.1021/ie202974u

Facco, P., Zomer, S., Rowland-Jones, R. C., Marsh, D., Diaz-Fernandez, P., Finka, G., Bezzo, F., and Barolo, M. (2020). Using data analytics to accelerate biopharmaceutical process scale-up. *Biochemical Engineering Journal*, **164**(September), 107791. https://doi.org/10.1016/j.bej.2020.107791

Fernandes, J. M. C., Fraga, I., Sousa, R. M. O. F., Rodrigues, M. A. M., Sampaio, A., Bezerra, R. M. F., and Dias, A. A. (2020). Pretreatment of Grape Stalks by Fungi: Effect on Bioactive Compounds, Fiber Composition, Saccharification Kinetics and

Monosaccharides Ratio. *International Journal of Environmental Research and Public Health*, **17**(16), 5900. https://doi.org/10.3390/ijerph17165900

Fernandes, N. C. P., Rato, T. J., and Reis, M. S. (2022). Modeling in the observable or latent space? A comparison of dynamic latent variable based monitoring methods for sensor fault detection. *Chemometrics and Intelligent Laboratory Systems*, **231**, 104684. https://doi.org/10.1016/j.chemolab.2022.104684

Ferrer, A. (2020). Discussion of "A review of data science in business and industry and a future view" by Grazia Vicario and Shirley Coleman. *Applied Stochastic Models in Business and Industry*, **36**(1), 23–29. https://doi.org/10.1002/asmb.2516

Ferrer, A. (2021). Multivariate six sigma: A key improvement strategy in industry 4.0. *Quality Engineering*, **33**(4), 758–763. https://doi.org/10.1080/08982112.2021.1957481

Feurer, M., Klein, A., Springenberg, J. T., Blum, M., and Hutter, F. (2015). Efficient and robust automated machine learning. *Proceeding of the 29th Annual Conference on Neural Information Processing Systems, NIPS 2015*, 2962–2970. https://doi.org/10.1007/978-3-030-05318-5_6

Filzmoser, P., Liebmann, B., and Varmuza, K. (2009). Repeated double cross validation. *Journal of Chemometrics*, **23**(4), 160–171. https://doi.org/10.1002/cem.1225

Flores-Cerrillo, J., and MacGregor, J. F. (2004). Control of batch product quality by trajectory manipulation using latent variable models. *Journal of Process Control*, **14**(5), 539–553. https://doi.org/10.1016/j.jprocont.2003.09.008

Flores-Cerrillo, J., and MacGregor, J. F. (2005). Latent variable MPC for trajectory tracking in batch processes. *Journal of Process Control*, **15**(6), 651–663. https://doi.org/10.1016/j.jprocont.2005.01.004

Forte, A., Zucaro, A., Basosi, R., and Fierro, A. (2016). LCA of 1,4-butanediol produced via direct fermentation of sugars from wheat straw feedstock within a territorial biorefinery. *Materials*, **9**(7), 1–22. https://doi.org/10.3390/MA9070563

Frank, L. E., and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, **35**(2), 109–135. https://doi.org/10.1080/00401706.1993.10485033

Fransson, M., and Folestad, S. (2006). Real-time alignment of batch process data using COW for on-line process monitoring. *Chemometrics and Intelligent Laboratory Systems*, **84**(1-2 SPEC. ISS.), 56–61. https://doi.org/10.1016/j.chemolab.2006.04.020

Gaffey, J., Rajauria, G., McMahon, H., Ravindran, R., Dominguez, C., Ambye-Jensen, M., Souza, M. F., Meers, E., Aragonés, M. M., Skunca, D., and Sanders, J. P. M. (2023). Green Biorefinery systems for the production of climate-smart sustainable products from grasses, legumes and green crop residues. *Biotechnology Advances*, **66**, 108168. https://doi.org/10.1016/j.biotechadv.2023.108168

García Muñoz, S. (2014). Two novel methods to analyze the combined effect of multiple raw-

materials and processing conditions on the product's final attributes: JRPLS and TPLS. *Chemometrics and Intelligent Laboratory Systems*, **133**, 49–62. https://doi.org/10.1016/j.chemolab.2014.02.006

García-Muñoz, S. (2004). *Batch Process Improvement using Latent Variable Methods* [McMaster University]. http://hdl.handle.net/11375/6274

García-Muñoz, S., Kourti, T., MacGregor, J. F., Apruzzese, F., and Champagne, M. (2006). Optimization of batch operating policies. Part I. Handling multiple solutions. *Industrial and Engineering Chemistry Research*, **45**(23), 7856–7866. https://doi.org/10.1021/ie060314g

García-Muñoz, S., Kourti, T., MacGregor, J. F., Mateos, A. G., and Murphy, G. (2003). Troubleshooting of an Industrial Batch Process Using Multivariate Methods. *Industrial & Engineering Chemistry Research*, **42**(15), 3592–3601. https://doi.org/10.1021/ie0300023

García-Muñoz, S., MacGregor, J. F., and Kourti, T. (2005). Product transfer between sites using Joint-Y PLS. *Chemometrics and Intelligent Laboratory Systems*, **79**(1–2), 101–114. https://doi.org/10.1016/j.chemolab.2005.04.009

García-Muñoz, S., MacGregor, J. F., Neogi, D., Latshaw, B. E., and Mehta, S. (2008). Optimization of batch operating policies. Part II. incorporating process constraints and industrial applications. *Industrial and Engineering Chemistry Research*, **47**(12), 4202–4208. https://doi.org/10.1021/ie071437j

García-Muñoz, S., Zhang, L., and Cortese, M. (2009). Root cause analysis during process development using Joint-Y PLS. *Chemometrics and Intelligent Laboratory Systems*, **95**(1), 101–105. https://doi.org/10.1016/j.chemolab.2008.09.006

Garud, S. S., Karimi, I. A., and Kraft, M. (2017). Design of computer experiments: A review. *Computers and Chemical Engineering*, **106**, 71–95. https://doi.org/10.1016/j.compchemeng.2017.05.010

Gavrilescu, M. (2014). Biorefinery Systems. In *Bioenergy Research: Advances and Applications* (pp. 219–241). Elsevier. https://doi.org/10.1016/B978-0-444-59561-4.00014-0

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, **70**(350), 320–328. https://doi.org/10.1080/01621459.1975.10479865

Geissler, S., Wintgens, T., Melin, T., Vossenkaul, K., and Kullmann, C. (2005). Modelling approaches for filtration processes with novel submerged capillary modules in membrane bioreactors for wastewater treatment. *Desalination*, **178**(1–3), 125–134. https://doi.org/10.1016/j.desal.2004.11.032

Geladi, P., and Kowalski, B. R. (1986). Partial Least-Squares Regression: A Tutorial. *Analytica Chimica Acta*, **185**, 1–17. https://doi.org/10.1016/0003-2670(86)80028-9

Gerardo, M. L., Oatley-Radcliffe, D. L., and Lovitt, R. W. (2014). Integration of membrane technology in microalgae biorefineries. *Journal of Membrane Science*, **464**, 86–99. https://doi.org/10.1016/j.memsci.2014.04.010

Ghosh, D., Mhaskar, P., and Macgregor, J. F. (2021). Hybrid Partial Least Squares Models for Batch Processes: Integrating Data with Process Knowledge. *Industrial & Engineering Chemistry Research*, **60**, 9508–9520. https://doi.org/10.1021/acs.iecr.1c00865

Goeman, J. J., and Solari, A. (2014). Multiple hypothesis testing in genomics. *Statistics in Medicine*, **33**(11), 1946–1978. https://doi.org/10.1002/sim.6082

Golshan, M., MacGregor, J. F., and Mhaskar, P. (2011). Latent variable model predictive control for trajectory tracking in batch processes: Alternative modeling approaches. *Journal of Process Control*, **21**(9), 1345–1358. https://doi.org/10.1016/j.jprocont.2011.06.007

Golub, G. H., and Van Loan, C. F. (2013). *Matrix Computations* (4th ed.). The Johns Hopkins University Press.

González Martínez, J. M., de Noord, O. E., and Ferrer, A. (2014a). Multisynchro: A novel approach for batch synchronization in scenarios of multiple asynchronisms. *Journal of Chemometrics*, **28**(5), 462–475. https://doi.org/10.1002/cem.2620

González Martínez, J. M., Vitale, R., de Noord, O. E., and Ferrer, A. (2014b). Effect of synchronization on bilinear batch process modeling. *Industrial and Engineering Chemistry Research*, **53**(11), 4339–4351. https://doi.org/10.1021/ie402052v

González-Martínez, J. M., Ferrer, A., and Westerhuis, J. A. (2011). Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping. *Chemometrics and Intelligent Laboratory Systems*, **105**(2), 195–206. https://doi.org/10.1016/j.chemolab.2011.01.003

Goodlin, B. E., Boning, D. S., Sawin, H. H., and Wise, B. M. (2003). Simultaneous Fault Detection and Classification for Semiconductor Manufacturing Tools. *Journal of The Electrochemical Society*, **150**(12), G778–G784. https://doi.org/10.1149/1.1623772

Gower, J. C., and Blasius, J. (2005). Multivariate Prediction with Nonlinear Principal Components Analysis: Theory. *Quality & Quantity*, **39**(4), 359–372. https://doi.org/10.1007/s11135-005-3005-1

Grand View Research. (2022). *1,4 Butanediol Market Size, Share & Trends Analysis Report By Application (Tetrahydrofuran, Polybutylene Terephthalate, Gamma-Butyrolactone), By Region, And Segment Forecasts, 2022—2030*. https://www.grandviewresearch.com/industry-analysis/1-4-butanediol-market

Grisales Díaz, V. H., Prado-Rubio, O. A., Willis, M. J., and Von Stosch, M. (2017). Dynamic hybrid model for ultrafiltration membrane processes. *Proceedings of the 27th European Symposium on Computer Aided Process Engineering – ESCAPE 27*, **1**, 193–198. https://doi.org/10.1016/B978-0-444-63965-3.50034-9

Guan, N., Li, J., Shin, H., Wu, J., Du, G., Shi, Z., Liu, L., and Chen, J. (2015). Comparative metabolomics analysis of the key metabolic nodes in propionic acid synthesis in Propionibacterium acidipropionici. *Metabolomics*, **11**(5), 1106–1116. https://doi.org/10.1007/s11306-014-0766-3

Gurden, S. P., Westerhuis, J. A., Bro, R., and Smilde, A. K. (2001). A comparison of multiway regression and scaling methods. *Chemometrics and Intelligent Laboratory Systems*, **59**(1–2), 121–136. https://doi.org/10.1016/S0169-7439(01)00168-X

H2O AI. (2023). *H2O AutoML*. https://github.com/h2oai/h2o-3

Han, H.-G., Zhang, H.-J., Liu, Z., and Qiao, J.-F. (2020). Data-driven decision-making for wastewater treatment process. *Control Engineering Practice*, **96**, 104305. https://doi.org/10.1016/j.conengprac.2020.104305

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, **16**(12), 2639–2664. https://doi.org/10.1162/0899766042321814

Hassani, S., Martens, H., Qannari, E. M., and Kohler, A. (2012). Degrees of freedom estimation in Principal Component Analysis and Consensus Principal Component Analysis. *Chemometrics and Intelligent Laboratory Systems*, **118**, 246–259. https://doi.org/10.1016/j.chemolab.2012.05.015

Hastie, T., and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, **84**(406), 502–516. https://doi.org/10.1080/01621459.1989.10478797

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.

Hatti-Kaul, R. (2010). Downstream Processing in Industrial Biotechnology. In *Industrial Biotechnology: Sustainable Growth and Economic Success* (pp. 279–321). https://doi.org/10.1002/9783527630233.ch8

He, Q. P., and Wang, J. (2011). Statistics Pattern Analysis: A New Process Monitoring Framework and its Application to Semiconductor Batch Processes. *AIChE Journal*, **57**(1), 107–121. https://doi.org/10.1002/aic.12247

He, Q. P., and Wang, J. (2018). Statistical process monitoring as a big data analytics tool for smart manufacturing. *Journal of Process Control*, **67**, 35–43. https://doi.org/10.1016/j.jprocont.2017.06.012

Helleckes, L. M., Hemmerich, J., Wiechert, W., von Lieres, E., and Grünberger, A. (2022). Machine learning in bioprocess development: From promise to practice. *Trends in Biotechnology*, S0167779922002815. https://doi.org/10.1016/j.tibtech.2022.10.010

Henneke, D., Hagedorn, A., Budman, H. M., and Legge, R. L. (2005). Application of spectrofluorometry to the prediction of PHB concentrations in a fed-batch process. *Bioprocess and Biosystems Engineering*, **27**(6), 359–364. https://doi.org/10.1007/s00449-004-0375-z

Henze, N., and Zirkler, B. (1990). A class of invariant consistent tests for multivariate normality. *Communications in Statistics - Theory and Methods*, **19**(10), 3595–3617. https://doi.org/10.1080/03610929008830400

Hochberg, Y. (1988). A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, **75**(4), 800–802. https://doi.org/10.1093/biomet/75.4.800

Holm-Nielsen, J. B., and Esbensen, K. H. (2011). Monitoring of biogas test plants-a process analytical technology approach. *Journal of Chemometrics*, **25**(7), 357–365. https://doi.org/10.1002/cem.1344

Hoskuldsson, A. (1988). PLS Regression Methods. *Journal of Chemometrics*, **2**, 211–228.

Hotelling, H. (1933). Analysis of a complex of statstical variables into principal components. *Journal of Educational Psychology*, **24**(6), 417–441. https://doi.org/10.1037/h0071325

Hotelling, H. (1936). Relations Between Two Sets of Variates. *Biometrika*, **28**(3/4), 321–377. https://doi.org/10.2307/2333955

Huang, W., Zhu, Y., Wang, L., Lv, W., Dong, B., and Zhou, W. (2021). Reversible and irreversible membrane fouling in hollow-fiber UF membranes filtering surface water: Effects of ozone/powdered activated carbon treatment. *RSC Advances*, **11**(17), 10323–10335. https://doi.org/10.1039/D0RA09820E

Hutter, F., Kotthoff, L., and Vanschoren, J. (2019). *Automated Machine Learning*. Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5

Hwang, T.-M., Oh, H., Choi, Y.-J., Nam, S.-H., Lee, S., and Choung, Y.-K. (2009). Development of a statistical and mathematical hybrid model to predict membrane fouling and performance. *Desalination*, **247**(1–3), 210–221. https://doi.org/10.1016/j.desal.2008.12.025

ICH. (2009). ICH Harmonised Tripartite Guideline, Guidance for Industry, Pharmaceutical Development Q8(R2). *International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use*.

IEA Bioenergy: Task 42 Biorefining in a circular economy, Annevelink, B., Garcia-Chavez, L., Van Ree, R., and Vural Gursel, I. (2022). *Global biorefinery status report 2022*. IEA Bioenergy. https://task42.ieabioenergy.com/publications/global-biorefinery-status-report-2022/

Ioannidou, S. M., Pateraki, C., Ladakis, D., Papapostolou, H., Tsakona, M., Vlysidis, A., Kookos, I. K., and Koutinas, A. (2020). Sustainable production of bio-based chemicals and polymers via integrated biomass refining and bioprocessing in a circular bioeconomy context. *Bioresource Technology*, **307**(March), 123093. https://doi.org/10.1016/j.biortech.2020.123093

Isermann, R. (1994). Integration of Fault Detection and Diagnosis Methods. *IFAC Proceedings Volumes*, **27**(5), 575–590. https://doi.org/10.1016/s1474-6670(17)48088-8

Isermann, R. (2005). Model-based fault-detection and diagnosis—Status and applications.

Annual Reviews in Control, **29**(1), 71–85. https://doi.org/10.1016/j.arcontrol.2004.12.002

Izenman, A. J. (2008). *Modern Multivariate Statistical Techniques* (1st ed.). Springer.

Jackson, J. E. (1959). Quality Control Methods for Several Related Variables. *Technometrics*, **1**(4), 359–377. https://doi.org/10.2307/1266717

Jackson, J. E. (1991). *A User's Guide To Princal Components* (1st ed.). Wiley.

Jackson, J. E., and Mudholkar, G. S. (1979). Control Procedures for Residuals Associated with Principal Component Analysis. *Technometrics*, **21**(3), 341–349. https://doi.org/10.2307/1267757

Jaeckle, C. M., and MacGregor, J. F. (1998). Product design through multivariate statistical analysis of process data. *AIChE Journal*, **44**(5), 1105–1118. https://doi.org/10.1002/aic.690440509

Jaeckle, C. M., and MacGregor, J. F. (2000). Industrial applications of product design through the inversion of latent variable models. *Chemometrics and Intelligent Laboratory Systems*, **50**(2), 199–210. https://doi.org/10.1016/S0169-7439(99)00058-1

Jia, F., Martin, E. B., and Morris, A. J. (2000). Non-linear principal components analysis with application to process fault detection. *International Journal of Systems Science*, **31**(11), 1473–1487. https://doi.org/10.1080/00207720050197848

Jia, Q., and Zhang, Y. (2016). Quality-related fault detection approach based on dynamic kernel partial least squares. *Chemical Engineering Research and Design*, **106**, 242–252. https://doi.org/10.1016/j.cherd.2015.12.015

Jiang, L. Y., and Zhu, J. M. (2013). Separation technologies for current and future biorefineries—Status and potential of membrane-based separation. *WIREs Energy and Environment*, **2**(6), 673–690. https://doi.org/10.1002/wene.73

Jiang, R., Linzon, Y., Vitkin, E., Yakhini, Z., Chudnovsky, A., and Golberg, A. (2016). Thermochemical hydrolysis of macroalgae Ulva for biorefinery: Taguchi robust design method. *Scientific Reports*, **6**(1), 27761. https://doi.org/10.1038/srep27761

Jiao, J., Yu, H., and Wang, G. (2015). A quality-related fault detection approach based on dynamic least squares for process monitoring. *IEEE Transactions on Industrial Electronics*, **63**(4), 2625–2632. https://doi.org/10.1109/TIE.2015.2497204

Jin, H., Song, Q., and Hu, X. (2019). Auto-Keras: An Efficient Neural Architecture Search System. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1946–1956. https://doi.org/10.1145/3292500.3330648

Jollife, I. T., and Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **374**(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Juang, R.-S., Chen, H.-L., and Chen, Y.-S. (2008). Membrane fouling and resistance analysis in dead-end ultrafiltration of Bacillus subtilis fermentation broths. *Separation and Purification Technology*, **63**(3), 531–538. https://doi.org/10.1016/j.seppur.2008.06.011

Julio, R., Albet, J., Vialle, C., Vaca-Garcia, C., and Sablayrolles, C. (2017). Sustainable design of biorefinery processes: Existing practices and new methodology. *Biofuels, Bioproducts and Biorefining*, **11**(2), 373–395. https://doi.org/10.1002/bbb.1749

Kadlec, P., Gabrys, B., and Strandt, S. (2009). Data-driven Soft Sensors in the process industry. *Computers & Chemical Engineering*, **33**(4), 795–814. https://doi.org/10.1016/j.compchemeng.2008.12.012

Kadlec, P., Grbić, R., and Gabrys, B. (2011). Review of adaptation mechanisms for data-driven soft sensors. *Computers and Chemical Engineering*, **35**(1), 1–24. https://doi.org/10.1016/j.compchemeng.2010.07.034

Kallioinen, M., Huuhilo, T., Reinikainen, S. P., Nuortila-Jokinen, J., and Mänttäri, M. (2006). Examination of membrane performance with multivariate methods: A case study within a pulp and paper mill filtration application. *Chemometrics and Intelligent Laboratory Systems*, **84**(1-2 SPEC. ISS.), 98–105. https://doi.org/10.1016/j.chemolab.2006.04.015

Kaneko, H., and Funatsu, K. (2013). A chemometric approach to prediction of transmembrane pressure in membrane bioreactors. *Chemometrics and Intelligent Laboratory Systems*, **126**, 30–37. https://doi.org/10.1016/j.chemolab.2013.04.016

Kang, K. H., Hong, U. G., Bang, Y., Choi, J. H., Kim, J. K., Lee, J. K., Han, S. J., and Song, I. K. (2015). Hydrogenation of succinic acid to 1,4-butanediol over Re–Ru bimetallic catalysts supported on mesoporous carbon. *Applied Catalysis A: General*, **490**, 153–162. https://doi.org/10.1016/j.apcata.2014.11.029

Kassidas, A., MacGregor, J. F., and Taylor, P. A. (1998). Synchronization of Batch Trajectories Using Dynamic Time Warping. *AIChE Journal*, **44**(4), 864–875. https://doi.org/10.1002/aic.690440412

Kim, J. H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics and Data Analysis*, **53**(11), 3735–3745. https://doi.org/10.1016/j.csda.2009.04.009

Kini, K. R., and Madakyaru, M. (2019). Anomaly detection using multi-scale dynamic principal component analysis for Tenneesse Eastman Process. *2019 5th Indian Control Conference, ICC 2019 - Proceedings*, 219–224. https://doi.org/10.1109/INDIANCC.2019.8715552

Klimkiewicz, A., Cervera-Padrell, A. E., and van den Berg, F. W. J. (2016). Multilevel Modeling for Data Mining of Downstream Bio-Industrial Processes. *Chemometrics and Intelligent Laboratory Systems*, **154**, 62–71. https://doi.org/10.1016/j.chemolab.2016.03.020

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model

selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, **2**, 1137–1143.

Korkmaz, S., Goksuluk, D., and Zararsiz, G. (2014). MVN: An R package for assessing multivariate normality. *R Journal*, **6**(2), 151–162. https://doi.org/10.32614/rj-2014-031

Kosanovich, K. A., Dahl, K. S., and Piovoso, M. J. (1996). Improved Process Understanding Using Multiway Principal Component Analysis. *Industrial and Engineering Chemistry Research*, **35**(1), 138–146. https://doi.org/10.1021/ie9502594

Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., and Leyton-Brown, K. (2017). Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA. *Journal of Machine Learning Research*, **18**, 1–5. https://doi.org/10.1007/978-3-030-05318-5_4

Kourti, T. (2003). Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, **17**(1), 93–109. https://doi.org/10.1002/cem.778

Kourti, T. (2019). Pharmaceutical manufacturing: The role of multivariate analysis in design space, control strategy, process understanding, troubleshooting, and optimization. In D. J. Am Ende and M. T. Am Ende (Eds.), *Chemical Engineering in the Pharmaceutical Industry* (pp. 601–629). John Wiley & Sons, Inc. https://doi.org/10.1002/9781119600800.ch75

Kourti, T., and MacGregor, J. F. (1995). Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, **28**(1), 3–21. https://doi.org/10.1016/0169-7439(95)80036-9

Kourti, T., and MacGregor, J. F. (1996). Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, **28**(4), 409–428. https://doi.org/10.1080/00224065.1996.11979699

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, **37**(2), 233–243. https://doi.org/10.1002/aic.690370209

Krämer, N., and Sugiyama, M. (2011). The degrees of freedom of partial least squares regression. *Journal of the American Statistical Association*, **106**(494), 697–705. https://doi.org/10.1198/jasa.2011.tm10107

Kresta, J. V., Macgregor, J. F., and Marlin, T. E. (1991). Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering*, **69**(1), 35–47. https://doi.org/10.1002/cjce.5450690105

Kroll, P., Stelzer, I. V., and Herwig, C. (2017). Soft sensor for monitoring biomass subpopulations in mammalian cell culture processes. *Biotechnology Letters*, **39**(11), 1667–1673. https://doi.org/10.1007/s10529-017-2408-0

Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory*

Systems, **30**(1), 179–196. https://doi.org/10.1016/0169-7439(95)00076-3

Kumar, V., Bhat, S. A., Kumar, S., Verma, P., Badruddin, I. A., Américo-Pinheiro, J. H. P., Sathyamurthy, R., and Atabani, A. E. (2023). Tea byproducts biorefinery for bioenergy recovery and value-added products development: A step towards environmental sustainability. *Fuel*, **350**, 128811. https://doi.org/10.1016/j.fuel.2023.128811

Kvalheim, O. M. (2010). Interpretation of partial least squares regression models by means of target projection and selectivity ratio plots. *Journal of Chemometrics*, **24**(7–8), 496–504. https://doi.org/10.1002/cem.1289

Lakshminarayanan, S., Gudi, R. D., Shah, S. L., and Nandakumar, K. (1996). Monitoring Batch Processes Using Multivariate Statistical Tools: Extensions and Practical Issues. *IFAC Proceedings Volumes*, **29**(1), 6037–6042. https://doi.org/10.1016/S1474-6670(17)58648-6

Lakshminarayanan, S., Patwardhan, R. S., Shah, S. L., and Nandakumar, K. (1997). A Dynamic PLS Framework for Constrained Model Predictive Control. *IFAC Proceedings Volumes*, **30**(9), 541–546. https://doi.org/10.1016/s1474-6670(17)43205-8

Lakshminarayanan, S., Shah, S. L., and Nandakumar, K. (1995). Identification of Hammerstein models using multivariate statistical tools. *Chemical Engineering Science*, **50**(22), 3599–3613. https://doi.org/10.1016/0009-2509(95)00182-5

Larimore, W. E. (1983). System Identification, Reduced-Order Filtering and Modeling via Canonical Variate Analysis. *Proceedings of the American Control Conference*, 445–451. https://doi.org/10.23919/ACC.1983.4788156

Larimore, W. E. (1990). Canonical variate analysis in identification, filtering, and adaptive control. *Proceedings of the IEEE Conference on Decision and Control*, **2**, 596–604. https://doi.org/10.1109/cdc.1990.203665

Larimore, W. E., and Baillieul, J. (1990). Identification and Filtering of Nonlinear Systems Using Canonical Variate Analysis. *Proceedings of the 29th Conference on Decision and Control*, 635–640. https://doi.org/10.1109/CDC.1987.272758

Le, T. T., Fu, W., and Moore, J. H. (2020). Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics*, **36**(1), 250–256. https://doi.org/10.1093/bioinformatics/btz470

Lee, D. S., Vanrolleghem, P. A., and Park, J. M. (2005). Parallel hybrid modeling methods for a full-scale cokes wastewater treatment plant. *Journal of Biotechnology*, **115**(3), 317–328. https://doi.org/10.1016/j.jbiotec.2004.09.001

Lee, J. M., Yoo, C. K., and Lee, I. B. (2004a). Enhanced process monitoring of fed-batch penicillin cultivation using time-varying and multivariate statistical analysis. *Journal of Biotechnology*, **110**(2), 119–136. https://doi.org/10.1016/j.jbiotec.2004.01.016

Lee, J. M., Yoo, C. K., and Lee, I. B. (2004b). Statistical process monitoring with independent component analysis. *Journal of Process Control*, **14**(5), 467–485.

https://doi.org/10.1016/j.jprocont.2003.09.004

Lee, L. C., Liong, C. Y., and Jemain, A. A. (2018). Partial least squares-discriminant analysis (PLS-DA) for classification of high-dimensional (HD) data: A review of contemporary practice strategies and knowledge gaps. *Analyst*, **143**(15), 3526–3539. https://doi.org/10.1039/c8an00599k

Lee, S. Y., Kim, H. U., Chae, T. U., Cho, J. S., Kim, J. W., Shin, J. H., Kim, D. I., Ko, Y.-S., Jang, W. D., and Jang, Y.-S. (2019). A comprehensive metabolic map for production of bio-based chemicals. *Nature Catalysis*, **2**(1), 18–33. https://doi.org/10.1038/s41929-018-0212-4

Lennox, B., Montague, G. A., Hiden, H. G., Kornfeld, G., and Goulding, P. R. (2001). Process monitoring of an industrial fed-batch fermentation. *Biotechnology and Bioengineering*, **74**(2), 125–135. https://doi.org/10.1002/bit.1102

Li, G., Liu, B., Qin, S. J., and Zhou, D. (2011). Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: The dynamic T-PLS approach. *IEEE Transactions on Neural Networks*, **22**(12 PART 2), 2262–2271. https://doi.org/10.1109/TNN.2011.2165853

Lin, B., Recke, B., Schmidt, T. M., Knudsen, J. K. H., and Jorgensen, S. B. (2009). Data-driven soft sensor design with multiple-rate sampled data: A comparative study. *Industrial & Engineering Chemistry Research*, **48**, 5379–5387. https://doi.org/10.1021/ie801084e

Linting, M., Meulman, J. J., Groenen, P. J. F., and van der Kooij, A. J. (2007). Nonlinear Principal Components Analysis: Introduction and Application. *Psychological Methods*, **12**(3), 336–358. https://doi.org/10.1037/1082-989X.12.3.336

Liu, Z., Bruwer, M. J., MacGregor, J. F., Rathore, S. S. S., Reed, D. E., and Champagne, M. J. (2011a). Modeling and optimization of a tablet manufacturing line. *Journal of Pharmaceutical Innovation*, **6**(3), 170–180. https://doi.org/10.1007/s12247-011-9112-8

Liu, Z., Bruwer, M. J., MacGregor, J. F., Rathore, S. S. S., Reed, D. E., and Champagne, M. J. (2011b). Scale-up of a pharmaceutical roller compaction process using a joint-Y partial least squares model. *Industrial and Engineering Chemistry Research*, **50**(18), 10696–10706. https://doi.org/10.1021/ie102316b

Ljung, G. M., and Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika*, **65**(2), 297–303. https://doi.org/10.1093/biomet/65.2.297

Ljung, L. (1999). *Systems Identification* (2nd ed.). Prentice Hall.

López-Garzón, C. S., and Straathof, A. J. J. (2014). Recovery of carboxylic acids produced by fermentation. *Biotechnology Advances*, **32**(5), 873–904. https://doi.org/10.1016/j.biotechadv.2014.04.002

Louwerse, D. J., Smilde, A. K., and Kiers, H. A. L. (1999a). Cross-validation of multiway component models. *Journal of Chemometrics*, **13**(5), 491–510.

https://doi.org/10.1002/(SICI)1099-128X(199909/10)13:5<491::AID-CEM537>3.0.CO;2-2

Louwerse, D. J., Tates, A. A., Smilde, A. K., Koot, G. L. M., and Berndt, H. (1999b). PLS discriminant analysis with contribution plots to determine differences between parallel batch reactors in the process industry. *Chemometrics and Intelligent Laboratory Systems*, **46**(2), 197–206. https://doi.org/10.1016/S0169-7439(98)00185-3

Lu, N., Yao, Y., Gao, F., and Wang, F. (2005). Two-dimensional dynamic PCA for batch process monitoring. *AIChE Journal*, **51**(12), 3300–3304. https://doi.org/10.1002/aic.10568

Luttmann, R., Bracewell, D. G., Cornelissen, G., Gernaey, K. V., Glassey, J., Hass, V. C., Kaiser, C., Preusse, C., Striedner, G., and Mandenius, C.-F. (2012). Soft sensors in bioprocessing: A status report and recommendations. *Biotechnology Journal*, **7**(8), 1040–1048. https://doi.org/10.1002/biot.201100506

Lv, F., Wen, C., Liu, M., and Bao, Z. (2018). Higher-order correlation-based multivariate statistical process monitoring. *Journal of Chemometrics*, **32**, e3033. https://doi.org/10.1002/cem.3033

MacGregor, J. F., and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, **3**(3), 403–414. https://doi.org/10.1016/0967-0661(95)00014-L

Maere, T., Villez, K., Marsili-Libelli, S., Naessens, W., and Nopens, I. (2012). Membrane bioreactor fouling behaviour assessment through principal component analysis and fuzzy clustering. *Water Research*, **46**(18), 6132–6142. https://doi.org/10.1016/j.watres.2012.08.027

Mainka, T., Mahler, N., Herwig, C., and Pflügl, S. (2019). Soft Sensor-Based Monitoring and Efficient Control Strategies of Biomass Concentration for Continuous Cultures of Haloferax mediterranei and Their Application to an Industrial Production Chain. *Microorganisms*, **7**(12), 648. https://doi.org/10.3390/microorganisms7120648

Makridakis, S. (1990). Sliding simulation: A new approach to time-series forecasting. *Management Science*, **36**(4), 505–512.

Malthouse, E. C., Tamhane, A. C., and Mah, R. S. H. (1997). Nonlinear partial least squares. *Computers and Chemical Engineering*, **21**(8), 875–890. https://doi.org/10.1016/S0098-1354(96)00311-0

Mancini, E., Mansouri, S. S., Gernaey, K. V., Luo, J., and Pinelo, M. (2020). From second generation feed-stocks to innovative fermentation and downstream techniques for succinic acid production. *Critical Reviews in Environmental Science and Technology*, **50**(18), 1829–1873. https://doi.org/10.1080/10643389.2019.1670530

Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, **57**(3), 519–530. https://doi.org/10.1093/biomet/57.3.519

Marques, R., von Stosch, M., Portela, R. M. C., Torres, C. A. V., Antunes, S., Freitas, F., Reis, M. A. M., and Oliveira, R. (2017). Hybrid modeling of microbial exopolysaccharide (EPS) production: The case of Enterobacter A47. *Journal of Biotechnology*, **246**, 61–70. https://doi.org/10.1016/j.jbiotec.2017.01.017

Martin, E. B., and Morris, A. J. (1996). Non-parametric confidence bounds for process performance monitoring charts. *Journal of Process Control*, **6**(6), 349–358. https://doi.org/10.1016/0959-1524(96)00010-8

Martin, M. A. (2010). First generation biofuels compete. *New Biotechnology*, **27**(5), 596–608. https://doi.org/10.1016/j.nbt.2010.06.010

Martín, M., and Grossmann, I. E. (2013). On the systematic synthesis of sustainable biorefineries. *Industrial and Engineering Chemistry Research*, **52**(9), 3044–3064. https://doi.org/10.1021/ie2030213

McCurdy, A. T., Higham, A. J., Morgan, M. R., Quinn, J. C., and Seefeldt, L. C. (2014). Two-step process for production of biodiesel blends from oleaginous yeast and microalgae. *Fuel*, **137**, 269–276. https://doi.org/10.1016/j.fuel.2014.07.099

McQuarrie, A. D. R., and Tsai, C.-L. (1998). *Regression and time series model selection* (1st ed.). World Scientific.

Mecklin, C. J., and Mundfrom, D. J. (2005). A Monte Carlo comparison of the Type I and Type II error rates of tests of multivariate normality. *Journal of Statistical Computation and Simulation*, **75**(2), 93–107. https://doi.org/10.1080/0094965042000193233

Meindersma, G. W., Augeraud, J., and Vergossen, F. H. P. (1997). Separation of a biocatalyst with ultrafiltration or filtration after bioconversion. *Journal of Membrane Science*, **125**(2), 333–349. https://doi.org/10.1016/S0376-7388(95)00081-X

Meneghetti, N., Barolo, M., and Tomba, E. (2012). *Trasferimento di prodotto tra apparecchiature diverse mediante tecniche statistiche multivariate: Applicazione a un processo di produzione di nanoparticelle* [Master's Thesis, University of Padova]. https://thesis.unipd.it/handle/20.500.12608/16376?1/Tesi_completa.pdf

MLJAR. (2023). *MLJAR*. https://github.com/mljar/mljar-supervised

Mohr, F., Arnese-Feffin, E., Barolo, M., and Braatz, R. D. (2023). *Smart process analytics for process monitoring* [In preparation].

Mohr, F., Sun, W., and Braatz, R. D. (2019, August 7). *Smart Process Data Analytics for Supervised Classification* [Poster Presentation]. Foundations of Process/Product Analytics and Machine Learning – FOPAM 2019, Raleigh (NC).

Monclús, H., Ferrero, G., Buttiglieri, G., Comas, J., and Rodriguez-Roda, I. (2011). Online monitoring of membrane fouling in submerged MBRs. *Desalination*, **277**(1–3), 414–419. https://doi.org/10.1016/j.desal.2011.04.055

Montgomery, D. C. (2009). *Introduction to Statistical Quality Control* (6th ed.). John Wiley & Sons, Inc.

Montgomery, D. C., and Runger, G. C. (2018). *Applied Statistics and Probability for Engineers* (7th ed.). Wiley.

Mortensen, P. P., and Bro, R. (2006). Real-time monitoring and chemical profiling of a cultivation process. *Chemometrics and Intelligent Laboratory Systems*, **84**(1–2), 106–113. https://doi.org/10.1016/j.chemolab.2006.04.022

Mulder, M. (1996). *Basic Principles of Membrane Technology* (2nd ed.). Kluwer Academic Publisher.

Müller, K. R., Mika, S., Rätsch, G., Tsuda, K., and Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, **12**(2), 181–201. https://doi.org/10.1201/9781420038613.ch4

Muteki, K., MacGregor, J. F., and Ueda, T. (2006). Rapid development of new polymer blends: The optimal selection of materials and blend ratios. *Industrial and Engineering Chemistry Research*, **45**(13), 4653–4660. https://doi.org/10.1021/ie050953b

Nachtergaele, P., Thybaut, J., De Meester, S., Drijvers, D., Saeys, W., and Dewulf, J. (2020). Multivariate Analysis of Industrial Biorefinery Processes: Strategy for Improved Process Understanding with Case Studies in Fatty Acid Production. *Industrial & Engineering Chemistry Research*, **59**(16), 7732–7745. https://doi.org/10.1021/acs.iecr.0c00515

Nadon, R., and Shoemaker, J. (2002). Statistical issues with microarrays: Processing and analysis. *TRENDS in Genetics*, **18**(5), 265–271. https://doi.org/10.1016/S0168-9525(02)02665-3

Naessens, W., Maere, T., Gilabert-Oriol, G., Garcia-Molina, V., and Nopens, I. (2017). PCA as tool for intelligent ultrafiltration for reverse osmosis seawater desalination pretreatment. *Desalination*, **419**(June), 188–196. https://doi.org/10.1016/j.desal.2017.06.018

Narayanan, H., Sokolov, M., Morbidelli, M., and Butté, A. (2019). A new generation of predictive models: The added value of hybrid models for manufacturing processes of therapeutic proteins. *Biotechnology and Bioengineering*, **116**(10), 2540–2549. https://doi.org/10.1002/bit.27097

Narayanan, H., Von Stosch, M., Feidl, F., Sokolov, M., Morbidelli, M., and Butté, A. (2023). Hybrid modeling for biopharmaceutical processes: Advantages, opportunities, and implementation. *Frontiers in Chemical Engineering*, **5**, 1157889. https://doi.org/10.3389/fceng.2023.1157889

Negiz, A., and Çlinar, A. (1997). Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE Journal*, **43**(8), 2002–2020. https://doi.org/10.1002/aic.690430810

Niitsuma, H., and Okada, T. (2005). Covariance and PCA for Categorical Variables. In T. B. Ho, D. Cheung, and H. Liu (Eds.), *Advances in Knowledge Discovery and Data Mining*

(Vol. 3518, pp. 523–528). Springer Berlin Heidelberg. https://doi.org/10.1007/11430919_61

Nomikos, P. (1996). Detection and diagnose of abnormal batch operations based on multi-way principal component analysis. *ISA Transactions*, **35**, 259–266.

Nomikos, P., and MacGregor, J. F. (1994). Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, **40**(8), 1361–1375. https://doi.org/10.1002/aic.690400809

Nomikos, P., and MacGregor, J. F. (1995a). Multivariate Processes SPC Charts for Monitoring Batch Processes. *Technometrics*, **37**(1), 41–59.

Nomikos, P., and MacGregor, J. F. (1995b). Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, **30**(1), 97–108. https://doi.org/10.1016/0169-7439(95)00043-7

Noorman, H. J., and Heijnen, J. J. (2017). Biochemical engineering's grand adventure. *Chemical Engineering Science*, **170**, 677–693. https://doi.org/10.1016/j.ces.2016.12.065

Nova Institute. (2017). *Biorefineries in Europe 2017*. https://task42.ieabioenergy.com/wp-content/uploads/sites/10/2018/05/MappingBiorefineriesAppendix_171219.pdf

Novamont S.p.A. (2016, September 29). *Opening of the world's first industrial scale plant for the production of butanediol via fermentation of renewable raw materials*. https://novamont.it/eng/read-press-release/mater-biotech/

Odiowei, P. P., and Cao, Y. (2009). Nonlinear Dynamic Process Monitoring using Canonical Variate Analysis and Kernel Density Estimations. *Proceedings of the 10th International Symposium on Process Systems Engineering*, 1557–1562. https://doi.org/10.1016/S1570-7946(09)70650-9

Odiowei, P. P., and Cao, Y. (2010). Nonlinear Dynamic Process Monitoring using Canonical Variate Analysis and Kernel Density Estimations. *IEEE Transactions on Industrial Informatics*, **6**(1), 36–45. https://doi.org/10.1109/TII.2009.2032654

O'Flaherty, R., Bergin, A., Flampouri, E., Mota, L. M., Obaidi, I., Quigley, A., Xie, Y., and Butler, M. (2020). Mammalian cell culture for production of recombinant proteins: A review of the critical steps in their biomanufacturing. *Biotechnology Advances*, **43**(May), 107552. https://doi.org/10.1016/j.biotechadv.2020.107552

Okada, T. (2000). A Note on Covariances for Categorical Data. In K. S. Leung, L.-W. Chan, and H. Meng (Eds.), *Intelligent Data Engineering and Automated Learning—IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents* (Vol. 1983, pp. 150–157). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44491-2_23

Oliveira, R. (2004). Combining first principles modelling and artificial neural networks: A general framework. *Computers and Chemical Engineering*, **28**(5), 755–766. https://doi.org/10.1016/j.compchemeng.2004.02.014

Oliveri, P., López, M. I., Casolino, M. C., Ruisánchez, I., Callao, M. P., Medini, L., and Lanteri, S. (2014). Partial least squares density modeling (PLS-DM)—A new class-modeling strategy applied to the authentication of olives in brine by near-infrared spectroscopy. *Analytica Chimica Acta*, **851**(C), 30–36. https://doi.org/10.1016/j.aca.2014.09.013

Oppong, F. B., and Agbedra, S. Y. (2016). Assessing Univariate and Multivariate Normality, A Guide For Non-Statisticians. *Mathematical Theory and Modeling*, **6**(2), 26–33.

Palací-López, D., Facco, P., Barolo, M., and Ferrer, A. (2019). New tools for the design and manufacturing of new products based on Latent Variable Model Inversion. *Chemometrics and Intelligent Laboratory Systems*, **194**(March), 103848. https://doi.org/10.1016/j.chemolab.2019.103848

Palací-López, D., Villalba, P., Facco, P., Barolo, M., and Ferrer, A. (2020). Improved formulation of the latent variable model inversion–based optimization problem for quality by design applications. *Journal of Chemometrics*, **34**(6), 1–18. https://doi.org/10.1002/cem.3230

Pallardy, R. (2023). *Deepwater Horizon oil spill*. Https://Www.Britannica.Com/Event/Deepwater-Horizon-Oil-Spill.

Paluš, M., and Dvořák, I. (1992). Singular-value decomposition in attractor reconstruction: Pitfalls and precautions. *Physica D: Nonlinear Phenomena*, **55**(1–2), 221–234. https://doi.org/10.1016/0167-2789(92)90198-V

Paredes, R., Rato, T. J., and Reis, M. S. (2023). Causal network inference and functional decomposition for decentralized statistical process monitoring: Detection and diagnosis. *Chemical Engineering Science*, **267**, 118338. https://doi.org/10.1016/j.ces.2022.118338

Parzen, E. (1962). On the Estimation of Probability Density Functions and Mode. *The Annals of Mathematical Statistics*, **33**(3), 1065–1076. https://doi.org/10.1214/aoms/1177704472

Pearson, K. (1901). On lines and places of closest fit to systems of points in space. *Philosophical Magazine*, **6**(2), 559–572. https://doi.org/10.1080/14786440109462720

Perera, Y. S., Ratnaweera, D. A. A. C., Dasanayaka, C. H., and Abeykoon, C. (2023). The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review. *Engineering Applications of Artificial Intelligence*, **121**, 105988. https://doi.org/10.1016/j.engappai.2023.105988

Perry, R., and Green, D. W. (2008). *Perry's Chemical Engineers' Handbook* (8th ed.). McGraw Hill.

Philippe, N., Stricker, A.-E., Racault, Y., Husson, A., Sperandio, M., and Vanrolleghem, P. (2013). Modelling the long-term evolution of permeability in a full-scale MBR: Statistical approaches. *Desalination*, **325**, 7–15. https://doi.org/10.1016/j.desal.2013.04.027

Pilario, K. E., Shafiee, M., Cao, Y., Lao, L., and Yang, S.-H. (2020). A Review of Kernel

Methods for Feature Extraction in Nonlinear Process Monitoring. *Processes*, **8**(1), 24. https://doi.org/10.3390/pr8010024

Piron, E., Latrille, E., and René, F. (1997). Application of artificial neural networks for crossflow microfiltration modelling: "Black-box" and semi-physical approaches. *Computers & Chemical Engineering*, **21**(9), 1021–1030. https://doi.org/10.1016/S0098-1354(96)00332-8

Pomerantsev, A. L., and Rodionova, O. Y. (2018). Multiclass partial least squares discriminant analysis: Taking the right way—A critical tutorial. *Journal of Chemometrics*, **32**(8), 1–16. https://doi.org/10.1002/cem.3030

Pomeroy, B., Grilc, M., and Likozar, B. (2022). Artificial neural networks for bio-based chemical production or biorefining: A review. *Renewable and Sustainable Energy Reviews*, **153**, 111748. https://doi.org/10.1016/j.rser.2021.111748

Pontius, K., Junicke, H., Gernaey, K. V., and Bevilacqua, M. (2020). Monitoring yeast fermentations by nonlinear infrared technology and chemometrics—Understanding process correlations and indirect predictions. *Applied Microbiology and Biotechnology*, **104**(12), 5315–5335. https://doi.org/10.1007/s00253-020-10604-0

Prochaska, K., Antczak, J., Regel-Rosocka, M., and Szczygiełda, M. (2018). Removal of succinic acid from fermentation broth by multistage process (membrane separation and reactive extraction). *Separation and Purification Technology*, **192**(July 2017), 360–368. https://doi.org/10.1016/j.seppur.2017.10.043

Prunescu, R. M. (2015). *Dynamic Modeling, Optimization, and Advanced Control for Large Scale Biorefineries*. Technical University of Denmark.

Psichogios, D. C., and Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, **38**(10), 1499–1511. https://doi.org/10.1002/aic.690381003

Python Software Foundation. (2022). *Python* (3.9.12) [Computer software]. Python Software Foundation. https://www.python.org

Qin, S. J. (2003). Statistical process monitoring: Basics and beyond. *Journal of Chemometrics*, **17**(8–9), 480–502. https://doi.org/10.1002/cem.800

Qin, S. J., and McAvoy, T. J. (1992). Nonlinear PLS modeling using neural networks. *Computers and Chemical Engineering Chemical Engineering*, **16**(4), 379–391. https://doi.org/10.1016/0098-1354(92)80055-E

Quenouille, M. H. (1949). Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, **11**(1), 68–84. https://doi.org/10.1111/j.2517-6161.1949.tb00023.x

R Foundation. (2022). *R* (4.2.0) [Computer software]. R Foundation. https://www.r-project.org

Raich, A., and Çinar, A. (1996). Statistical Process Monitoring and Disturbance Diagnosis in Multivariable Continuous Processes. *AIChE Journal*, **42**(4), 995–1009.

https://doi.org/10.1002/aic.690420412

Raissi, M., Perdikaris, P., and Karniadakis, G. E. M. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, **378**, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Rajulapati, L., Chinta, S., Shyamala, B., and Rengaswamy, R. (2022). Integration of machine learning and first principles models. *AIChE Journal*, **68**(6), e17715. https://doi.org/10.1002/aic.17715

Ramaker, H. J., Van Sprang, E. N. M., Westerhuis, J. A., Gurden, S. P., and Smilde, A. K. (2006). Performance assessment and improvement of control charts for statistical batch process monitoring. *Statistica Neerlandica*, **60**(3), 339–360. https://doi.org/10.1111/j.1467-9574.2006.00337.x

Rao, C. R., and Mitra, S. K. (1971). Generalized Inverse of Matrices and Its Applications. In *Generalized Inverse of Matrices and Its Applications* (pp. 601–620). https://doi.org/10.2307/1266840

Rato, T. J., Blue, J., Pinaton, J., and Reis, M. S. (2017). Translation-Invariant Multiscale Energy-Based PCA for Monitoring Batch Processes in Semiconductor Manufacturing. *IEEE Transactions on Automation Science and Engineering*, **14**(2), 894–904. https://doi.org/10.1109/TASE.2016.2545744

Rato, T. J., and Reis, M. S. (2013). Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR). *Chemometrics and Intelligent Laboratory Systems*, **125**, 101–108. https://doi.org/10.1016/j.chemolab.2013.04.002

Reis, M. S., and Gins, G. (2017). Industrial process monitoring in the big data/industry 4.0 era: From detection, to diagnosis, to prognosis. *Processes*, **5**(3), 35. https://doi.org/10.3390/pr5030035

Reis, M. S., Gins, G., and Rato, T. J. (2019). Incorporation of process-specific structure in statistical process monitoring: A review. *Journal of Quality Technology*, **51**(4), 407–421. https://doi.org/10.1080/00224065.2019.1569954

Reis, M. S., and Kenett, R. (2018). Assessing the value of information of data-centric activities in the chemical processing industry 4.0. *AIChE Journal*, **64**(11), 3868–3881. https://doi.org/10.1002/aic.16203

Reis, M. S., and Rato, T. J. (2023). Hybrid modelling through latent differential-regression analysis (LDRA) for predicting long-term equipment degradation in the chemical process industry. *Chemical Engineering Science*, **278**, 118902. https://doi.org/10.1016/j.ces.2023.118902

Reis, M. S., Rendall, R., Rato, T. J., Martins, C., and Delgado, P. (2021a). Improving the sensitivity of statistical process monitoring of manifolds embedded in high-dimensional

spaces: The truncated-Q statistic. *Chemometrics and Intelligent Laboratory Systems*, **215**(March), 104369. https://doi.org/10.1016/j.chemolab.2021.104369

Reis, M. S., and Saraiva, P. M. (2021b). Data-centric process systems engineering: A push towards PSE 4.0. *Computers & Chemical Engineering*, **155**, 107529. https://doi.org/10.1016/j.compchemeng.2021.107529

Reis, M. S., and Saraiva, P. M. (2022). Data-Driven Process System Engineering–Contributions to its consolidation following the path laid down by George Stephanopoulos. *Computers and Chemical Engineering*, **159**, 107675. https://doi.org/10.1016/j.compchemeng.2022.107675

Rendall, R., Chiang, L. H., and Reis, M. S. (2019). Data-driven methods for batch data analysis – A critical overview and mapping on the complexity scale. *Computers and Chemical Engineering*, **124**, 1–13. https://doi.org/10.1016/j.compchemeng.2019.01.014

Rendall, R., Lu, B., Castillo, I., Chin, S. T., Chiang, L. H., and Reis, M. S. (2017a). A Unifying and Integrated Framework for Feature Oriented Analysis of Batch Processes. *Industrial and Engineering Chemistry Research*, **56**(30), 8590–8605. https://doi.org/10.1021/acs.iecr.6b04553

Rendall, R., Pereira, A. C., and Reis, M. S. (2017b). Advanced predictive methods for wine age prediction: Part I – A comparison study of single-block regression approaches based on variable selection, penalized regression, latent variables and tree-based ensemble methods. *Talanta*, **171**, 341–350. https://doi.org/10.1016/j.talanta.2016.10.062

Rényi, A. (1959). On Measures of Dependence. *Acta Mathematica Hungarica*, **10**(3–4), 441–451. https://doi.org/10.1007/BF02024507

Ricker, N. L. (1988). The use of biased least-squares estimators for parameters in discrete-time pulse-response models. *Industrial and Engineering Chemistry Research*, **27**(2), 343–350. https://doi.org/10.1021/ie00074a023

Rodionova, O. Y., Oliveri, P., and Pomerantsev, A. L. (2016). Rigorous and compliant approaches to one-class classification. *Chemometrics and Intelligent Laboratory Systems*, **159**(September), 89–96. https://doi.org/10.1016/j.chemolab.2016.10.002

Rosales-Calderon, O., and Arantes, V. (2019). A review on commercial-scale high-value products that can be produced alongside cellulosic ethanol. *Biotechnology for Biofuels*, **12**(1), 240. https://doi.org/10.1186/s13068-019-1529-1

Rosenblatt, M. (1956). Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, **27**(3), 832–837. https://doi.org/10.1214/aoms/1177728190

Rosipal, R. (2010). Nonlinear partial least squares: An overview. In *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques* (pp. 169–189). https://doi.org/10.4018/978-1-61520-911-8.ch009

Rosipal, R., and Trejo, L. J. (2001). Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. *Journal of Machine Learning Research*, **2**, 97–123. https://doi.org/10.1162/15324430260185556

Rothwell, S. G., Martin, E. B., and Morris, A. J. (1998). Comparison of Methods for Dealing with Uneven Length Batches. *IFAC Proceedings Volumes*, **31**(8), 387–392. https://doi.org/10.1016/S1474-6670(17)40216-3

Rousseeuw, P. J., and Van Zomeren, B. C. (1990). Unmasking Multivariate Outliers and Leverage Points. *Journal of the American Statistical Association*, **85**(411), 633–639. https://doi.org/10.1080/01621459.1990.10474920

Royston, J. P. (1983). Some Techniques for Assessing Multivariate Normality Based on the Shapiro-Wilk w. *Applied Statistics*, **32**(2), 121–133. https://doi.org/10.2307/2347291

rpy2. (2022). *Rpy2* (3.5.1) [Computer software]. https://github.com/rpy2/rpy2

Rudnitskaya, A., Costa, A. M. S., and Delgadillo, I. (2017). Calibration update strategies for an array of potentiometric chemical sensors. *Sensors and Actuators B: Chemical*, **238**, 1181–1189. https://doi.org/10.1016/j.snb.2016.06.075

Rudolph, G., Virtanen, T., Ferrando, M., Güell, C., Lipnizki, F., and Kallioinen, M. (2019). A review of in situ real-time monitoring techniques for membrane fouling in the biotechnology, biorefinery and food sectors. *Journal of Membrane Science*, **588**, 117221. https://doi.org/10.1016/j.memsci.2019.117221

Ruiz, S., Ortiz, M. C., Sarabia, L. A., and Sánchez, M. S. (2018). A computational approach to partial least squares model inversion in the framework of the process analytical technology and quality by design initiatives. *Chemometrics and Intelligent Laboratory Systems*, **182**(July), 70–78. https://doi.org/10.1016/j.chemolab.2018.08.014

Ruiz-García, A., and Nuez, I. (2016). Long-term performance decline in a brackish water reverse osmosis desalination plant. Predictive model for the water permeability coefficient. *Desalination*, **397**, 101–107. https://doi.org/10.1016/j.desal.2016.06.027

Russell, E. L., Chiang, L. H., and Braatz, R. D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, **51**(1), 81–93. https://doi.org/10.1016/S0169-7439(00)00058-7

Russell, S. A., Kesavan, P., Lee, J. H., and Ogunnaike, B. A. (1998). Recursive data-based prediction and control of product quality. *AIChE Journal*, **44**(11), 2442–2458. https://doi.org/10.1002/aic.690441112

Sá, M., Monte, J., Brazinha, C., Galinha, C. F., and Crespo, J. G. (2017). 2D Fluorescence spectroscopy for monitoring Dunaliella salina concentration and integrity during membrane harvesting. *Algal Research*, **24**, 325–332. https://doi.org/10.1016/j.algal.2017.04.013

Saccenti, E., and Camacho, J. (2015a). Determining the number of components in principal

components analysis: A comparison of statistical, crossvalidation and approximated methods. *Chemometrics and Intelligent Laboratory Systems*, **149**, 99–116. https://doi.org/10.1016/j.chemolab.2015.10.006

Saccenti, E., and Camacho, J. (2015b). On the use of the observation-wise k-fold operation in PCA cross-validation. *Journal of Chemometrics*, **29**(8), 467–478. https://doi.org/10.1002/cem.2726

Saha, K., R, U. M., Sikder, J., Chakraborty, S., Da Silva, S. S., and Dos Santos, J. C. (2017). Membranes as a tool to support biorefineries: Applications in enzymatic hydrolysis, fermentation and dehydration for bioethanol production. *Renewable and Sustainable Energy Reviews*, **74**, 873–890. https://doi.org/10.1016/j.rser.2017.03.015

Saleh, J. H., Haga, R. A., Favarò, F. M., and Bakolas, E. (2014). Texas City refinery accident: Case study in breakdown of defense-in-depth and violation of the safety–diagnosability principle in design. *Engineering Failure Analysis*, **36**, 121–133. https://doi.org/10.1016/j.engfailanal.2013.09.014

Salesforce. (2021). *TransmogrifAI*. https://github.com/salesforce/TransmogrifAI

Samuel, R. T., and Cao, Y. (2015a). Improved kernel canonical variate analysis for process monitoring. *2015 21st International Conference on Automation and Computing: Automation, Computing and Manufacturing for New Economic Growth, ICAC 2015*, 7313990. https://doi.org/10.1109/IConAC.2015.7313990

Samuel, R. T., and Cao, Y. (2015b). Kernel Canonical Variate Analysis for Nonlinear Dynamic Process Monitoring. *IFAC-PapersOnLine*, **48**(8), 605–610. https://doi.org/10.1016/j.ifacol.2015.09.034

Samuel, R. T., and Cao, Y. (2016). Nonlinear process fault detection and identification using kernel PCA and kernel density estimation. *Systems Science and Control Engineering*, **4**(1), 165–174. https://doi.org/10.1080/21642583.2016.1198940

Sansana, J., Joswiak, M. N., Castillo, I., Wang, Z., Rendall, R., Chiang, L. H., and Reis, M. S. (2021). Recent trends on hybrid modeling for Industry 4.0. *Computers and Chemical Engineering*, **151**, 107365. https://doi.org/10.1016/j.compchemeng.2021.107365

Satam, C. C., Daub, M., and Realff, M. J. (2019). Techno-economic analysis of 1,4-butanediol production by a single-step bioconversion process. *Biofuels, Bioproducts and Biorefining*, **13**(5), 1261–1273. https://doi.org/10.1002/bbb.2016

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Muller, K. R., Ratsch, G., and Smola, A. J. (1999). Input Space Versus Feature Space in Kernel-Based Methods. *IEEE Transactions on Neural Networks*, **10**(5), 1000–1017. https://doi.org/10.1109/72.788641

Schölkopf, B., Smola, A., and Müller, K. R. (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, **10**(5), 1299–1319. https://doi.org/10.1162/089976698300017467

Schubert, J., Simutis, R., Dors, M., Havlik, I., and Lubbert, A. (1994). Hybrid Modelling of Yeast Production Process. *Chemical Engineering Technology*, **17**, 10–20. https://doi.org/10.1002/ceat.270170103

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization* (1st ed.). Wiley.

Severson, K. A., Attia, P. M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M. H., Aykol, M., Herring, P. K., Fraggedakis, D., Bazant, M. Z., Harris, S. J., Chueh, W. C., and Braatz, R. D. (2019). Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, **4**(5), 383–391. https://doi.org/10.1038/s41560-019-0356-8

Severson, K., Chaiwatanodom, P., and Braatz, R. D. (2016). Perspectives on process monitoring of industrial systems. *Annual Reviews in Control*, **42**, 190–200. https://doi.org/10.1016/j.arcontrol.2016.09.001

Sharper, C. D., Larimore, W. E., Seborg, D. A., and Mellichamp, D. A. (1994). Identification of Chemical Processes Using Canonical Variate Analysis. *Computers & Chemical Engineering*, **18**(1), 55–69. https://doi.org/10.1016/0098-1354(94)85023-2

Shen, F., Ye, L., Ma, X., Ge, Z., and Song, Z. (2018). Multi-layer monitoring for parallel batch processes with input trajectory adjustment. *Proceedings of 2018 IEEE 7th Data Driven Control and Learning Systems Conference, DDCLS 2018*, 123–128. https://doi.org/10.1109/DDCLS.2018.8516026

Shewart, W. A. (1931). *Economic Control of Quality of Manufactured Product*. Quality Press.

Shi, X., Tal, G., Hankins, N. P., and Gitis, V. (2014). Fouling and cleaning of ultrafiltration membranes: A review. *Journal of Water Process Engineering*, **1**, 121–138. https://doi.org/10.1016/j.jwpe.2014.04.003

Shimizu, Y., Shimodera, K.-I., and Watanabe, A. (1993). Cross-flow microfiltration of bacterial cells. *Journal of Fermentation and Bioengineering*, **76**(6), 493–500. https://doi.org/10.1016/0922-338X(93)90247-6

Sikdar, S. K. (2003). Sustainable development and sustainability metrics. *AIChE Journal*, **49**(8), 1928–1932. https://doi.org/10.1002/aic.690490802

Silva, R. G. C., Ferreira, T. F., and Borges, É. R. (2020). Identification of potential technologies for 1,4-Butanediol production using prospecting methodology. *Journal of Chemical Technology and Biotechnology*, **95**(12), 3057–3070. https://doi.org/10.1002/jctb.6518

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis* (1st ed.). Chapman and Hall.

Solle, D., Hitzmann, B., Herwig, C., Pereira Remelhe, M., Ulonska, S., Wuerth, L., Prata, A., and Steckenreiter, T. (2017). Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation. *Chemie-Ingenieur-Technik*, **89**(5), 542–561. https://doi.org/10.1002/cite.201600175

Souza, F. A. A., Araújo, R., and Mendes, J. (2016). Review of soft sensor methods for

regression applications. *Chemometrics and Intelligent Laboratory Systems*, **152**, 69–79. https://doi.org/10.1016/j.chemolab.2015.12.011

Spiegler, K. S., and Kedem, O. (1966). Thermodynamics of hyperfiltration (reverse osmosis): Criteria for efficient membranes. *Desalination*, **1**(4), 311–326. https://doi.org/10.1016/S0011-9164(00)80018-1

Stephanopoulos, G., Locher, G., Duff, M. J., Kamimura, R., and Stephanopoulos, G. (1997). Fermentation database mining by pattern recognition. *Biotechnology and Bioengineering*, **53**(5), 443–452. https://doi.org/10.1002/(SICI)1097-0290(19970305)53:5<443::AID-BIT1>3.0.CO;2-H

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, **36**(2), 111–133. https://doi.org/10.1111/j.2517-6161.1974.tb00994.x

Strieder, M. M., Velásquez Piñas, J. A., Ampese, L. C., Costa, J. M., Carneiro, T. F., and Rostagno, M. A. (2023). Coffee biorefinery: The main trends associated with recovering valuable compounds from solid coffee residues. *Journal of Cleaner Production*, **415**, 137716. https://doi.org/10.1016/j.jclepro.2023.137716

Sun, W. (2020a). *Advanced Process Data Analytics* [Massachusetts Institute of Technology]. https://hdl.handle.net/1721.1/127569

Sun, W. (2020b). *Smart Process Analytics* (1.0) [Computer software]. Massachusetts Institute of Technology. https://github.com/vickysun5/SmartProcessAnalytics

Sun, W., and Braatz, R. D. (2021). Smart process analytics for predictive modeling. *Computers and Chemical Engineering*, **144**, 107134. https://doi.org/10.1016/j.compchemeng.2020.107134

Tan, S., and Mayrovouniotis, M. L. (1995). Reducing data dimensionality through optimizing neural network inputs. *AIChE Journal*, **41**(6), 1471–1480. https://doi.org/10.1002/aic.690410612

Tax, D. M. J., and Duin, R. P. W. (1999). Support vector domain description. *Pattern Recognition Letters*, **20**(11–13), 1191–1199. https://doi.org/10.1016/S0167-8655(99)00087-2

Tax, D. M. J., and Duin, R. P. W. (2004). Support Vector Data Description. *Machine Learning*, **54**, 45–66. https://doi.org/10.1023/B:MACH.0000008084.60811.49

Taylor, G. (2008). Biofuels and the biorefinery concept. *Energy Policy*, **36**(12), 4406–4409. https://doi.org/10.1016/j.enpol.2008.09.069

Teh, Chua, Hong, Ling, Andiappan, Foo, Hassim, and Ng. (2019). A Hybrid Multi-Objective Optimization Framework for Preliminary Process Design Based on Health, Safety and Environmental Impact. *Processes*, **7**(4), 200. https://doi.org/10.3390/pr7040200

Teixeira, A. P., Cunha, A. E., Clemente, J. J., Moreira, J. L., Cruz, H. J., Alves, P. M., Carrondo, M. J. T., and Oliveira, R. (2005). Modelling and optimization of a recombinant BHK-

21 cultivation process using hybrid grey-box systems. *Journal of Biotechnology*, **118**(3), 290–303. https://doi.org/10.1016/j.jbiotec.2005.04.024

Tessier, J., Duchesne, C., Tarcy, G. P., Gauthier, C., and Dufour, G. (2012). Multivariate analysis and monitoring of the performance of aluminum reduction cells. *Industrial and Engineering Chemistry Research*, **51**(3), 1311–1323. https://doi.org/10.1021/ie201258b

The Mathworks, Inc. (2021). *MATLAB* (Version R2021a) [Computer software]. The Mathworks, Inc. https://mathworks.com/products/matlab.html

The Mathworks, Inc. (2022a). *MATLAB* (Version R2022a) [Computer software]. The Mathworks, Inc. https://mathworks.com/products/matlab.html

The Mathworks, Inc. (2022b). *Systems Identification Toolbox* (9.16) [Computer software]. The Mathworks, Inc. https://mathworks.com/products/sysid.html

Thissen, U., Melssen, W. J., and Buydens, L. M. C. (2001). Nonlinear process monitoring using bottle-neck neural networks. *Analytica Chimica Acta*, **446**(1–2), 369–381. https://doi.org/10.1016/s0003-2670(01)01266-1

Tibshirani, R. (1988). Estimating Transformations for Regression Via Additivity and Variance Stabilization. *Journal of the America Statistical Association*, **83**(402), 394--405. https://doi.org/10.1080/01621459.1988.10478610

Tien, D. X., Khiang-Wee Lim, and Liu Jun. (2004). Comparative study of PCA approaches in process monitoring and fault detection. *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, **3**, 2594–2599. https://doi.org/10.1109/IECON.2004.1432212

Tomba, E., Barolo, M., and García Muñoz, S. (2012a). General framework for latent variable model inversion for the design and manufacturing of new products. *Industrial and Engineering Chemistry Research*, **51**(39), 12886–12900. https://doi.org/10.1021/ie301214c

Tomba, E., De Martin, M., Facco, P., Robertson, J., Zomer, S., Bezzo, F., and Barolo, M. (2013a). General procedure to aid the development of continuous pharmaceutical processes using multivariate statistical modeling-An industrial case study. *International Journal of Pharmaceutics*, **444**(1–2), 25–39. https://doi.org/10.1016/j.ijpharm.2013.01.018

Tomba, E., Facco, P., Bezzo, F., and García Muñoz, S. (2013b). Exploiting historical databases to design the target quality profile for a new product. *Industrial and Engineering Chemistry Research*, **52**(24), 8260–8271. https://doi.org/10.1021/ie3032839

Tomba, E., Facco, P., Bezzo, F., García Muñoz, S., and Barolo, M. (2012b). Combining fundamental knowledge and latent variable techniques to transfer process monitoring models between plants. *Chemometrics and Intelligent Laboratory Systems*, **116**, 67–77. https://doi.org/10.1016/j.chemolab.2012.04.016

Tomba, E., Meneghetti, N., Facco, P., Zelenková, T., Barresi, A. A., Marchisio, D. L., Bezzo, F., and Barolo, M. (2014). Transfer of a Nanoparticle Product Between Different Mixers Using Latent Variable Model Inversion. *AIChE Journal*, **60**(1), 123–135. https://doi.org/10.1002/aic.14244

Tracy, N. D., Young, J. C., and Mason, R. L. (1992). Multivariate Control Charts for Individual Observations. *Journal of Quality Technology*, **24**(2), 88–95. https://doi.org/10.1080/00224065.1992.12015232

Ubando, A. T., Felix, C. B., and Chen, W.-H. (2020). Biorefineries in circular bioeconomy: A comprehensive review. *Bioresource Technology*, **299**, 122585. https://doi.org/10.1016/j.biortech.2019.122585

Uurtio, V., Monteiro, J. M., Kandola, J., Shawe-Taylor, J., Fernandez-Reyes, D., and Rousu, J. (2018). A Tutorial on Canonical Correlation Methods. *ACM Computing Surveys*, **50**(6), 1–33. https://doi.org/10.1145/3136624

Valle, S., Li, W., and Qin, S. J. (1999). Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Industrial and Engineering Chemistry Research*, **38**(11), 4389–4401. https://doi.org/10.1021/ie990110i

Van Der Voet, H. (1999). Pseudo-degrees of freedom for complex predictive models: The example of partial least squares. *Journal of Chemometrics*, **13**(3–4), 195–208. https://doi.org/10.1002/(SICI)1099-128X(199905/08)13:3/4<195::AID-CEM540>3.0.CO;2-L

Vande Wouwer, A., Renotte, C., and Bogaerts, Ph. (2004). Biological reaction modeling using radial basis function networks. *Computers and Chemical Engineering*, **28**(11), 2157–2164. https://doi.org/10.1016/j.compchemeng.2004.03.003

Vanhatalo, E., and Kulahci, M. (2016). Impact of autocorrelation on principal components and their use in statistical process control. *Quality and Reliability Engineering International*, **32**(4), 1483–1500. https://doi.org/10.1002/qre.1858

Varma, S., and Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, **7**, 1–8. https://doi.org/10.1186/1471-2105-7-91

Varriale, L., and Ulber, R. (2023). Fungal-Based Biorefinery: From Renewable Resources to Organic Acids. *ChemBioEng Reviews*, **10**(3), 272–292. https://doi.org/10.1002/cben.202200059

Vela, M. C. V., Blanco, S. Á., García, J. L., and Rodríguez, E. B. (2008). Analysis of membrane pore blocking models applied to the ultrafiltration of PEG. *Separation and Purification Technology*, **62**(3), 489–498. https://doi.org/10.1016/j.seppur.2008.02.028

Velidandi, A., Kumar Gandam, P., Latha Chinta, M., Konakanchi, S., reddy Bhavanam, A., Raju Baadhe, R., Sharma, M., Gaffey, J., Nguyen, Q. D., and Gupta, V. K. (2023). State-of-the-art and future directions of machine learning for biomass characterization and for

sustainable biorefinery. *Journal of Energy Chemistry*, **81**, 42–63. https://doi.org/10.1016/j.jechem.2023.02.020

Venkatasubramanian, V. (2019). The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, **65**(2), 466–478. https://doi.org/10.1002/aic.16489

Vilker, V. L., Colton, C. K., Smith, K. A., and Green, D. L. (1984). The osmotic pressure of concentrated protein and lipoprotein solutions and its significance to ultrafiltration. *Journal of Membrane Science*, **20**(1), 63–77. https://doi.org/10.1016/S0376-7388(00)80723-1

Vitale, R., de Noord, O. E., Westerhuis, J. A., Smilde, A. K., and Ferrer, A. (2021). Divide et impera: How disentangling common and distinctive variability in multiset data analysis can aid industrial process troubleshooting and understanding. *Journal of Chemometrics*, **35**(2), 1–12. https://doi.org/10.1002/cem.3266

Vitale, R., Westerhuis, J. A., Naes, T., Smilde, A. K., de Noord, O. E., and Ferrer, A. (2017). Selecting the number of factors in principal component analysis by permutation testing-Numerical and practical aspects: Permutation testing in Principal Component Analysis. *Journal of Chemometrics*, **31**(12), e2937. https://doi.org/10.1002/cem.2937

von Stosch, M., Hamelink, J. M., and Oliveira, R. (2016). Hybrid modeling as a QbD/PAT tool in process development: An industrial E. coli case study. *Bioprocess and Biosystems Engineering*, **39**(5), 773–784. https://doi.org/10.1007/s00449-016-1557-1

von Stosch, M., Oliveira, R., Peres, J., and Feyo De Azevedo, S. (2011). A novel identification method for hybrid (N)PLS dynamical systems with application to bioprocesses. *Expert Systems with Applications*, **38**(9), 10862–10874. https://doi.org/10.1016/j.eswa.2011.02.117

von Stosch, M., Oliveira, R., Peres, J., and Feyo de Azevedo, S. (2014). Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers and Chemical Engineering*, **60**, 86–101. https://doi.org/10.1016/j.compchemeng.2013.08.008

Wang, C., Li, Q., Tang, H., Yan, D., Zhou, W., Xing, J., and Wan, Y. (2012). Membrane fouling mechanism in ultrafiltration of succinic acid fermentation broth. *Bioresource Technology*, **116**, 366–371. https://doi.org/10.1016/j.biortech.2012.03.099

Wang, H., and Yao, M. (2015). Chemometrics and Intelligent Laboratory Systems Fault detection of batch processes based on multivariate functional kernel principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, **149**, 78–89. https://doi.org/10.1016/j.chemolab.2015.09.018

Wang, J., Chen, X., Nan, Y., Zhou, J., and Xue, T. (2020). Narrow operating space based on the inversion of latent structures model for glycosylation process. *IEEE Access*, **8**, 190504–190515. https://doi.org/10.1109/ACCESS.2020.3031353

Wang, J., and He, Q. P. (2010). Multivariate Statistical Process Monitoring Based on Statistics

Pattern Analysis. *Industrial & Engineering Chemistry Research*, **49**(17), 7858–7869. https://doi.org/10.1021/ie901911p

Wang, L., and Shi, H. (2014). Improved Kernel PLS-based Fault Detection Approach for Nonlinear Chemical Processes. *Chinese Journal of Chemical Engineering*, **22**(6), 657–663. https://doi.org/10.1016/S1004-9541(14)60088-4

Wankat, P. C. (2009). *Separation Process Engineering* (4th ed.). Prentice Hall.

Wehrs, M., Tanjore, D., Eng, T., Lievense, J., Pray, T. R., and Mukhopadhyay, A. (2019). Engineering Robust Production Microbes for Large-Scale Cultivation. *Trends in Microbiology*, **27**(6), 524–537. https://doi.org/10.1016/j.tim.2019.01.006

Westerhuis, J. A., Kourti, T., and MacGregor, J. F. (1999). Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, **13**(3–4), 397–413. https://doi.org/10.1002/(sici)1099-128x(199905/08)13:3/4<397::aid-cem559>3.0.co;2-i

Whitaker, S. (1986). Flow in porous media I: A theoretical derivation of Darcy's law. *Transport in Porous Media*, **1**(1), 3–25. https://doi.org/10.1007/BF01036523

Wilson, D. J. H., Irwin, G. W., and Lightbody, G. (1997). Nonlinear PLS using Radial Basis Functions. *Proceedings of the American Control Conference*, 3275–3276. https://doi.org/10.1109/ACC.1997.612069

Wise, B. M., and Gallagher, N. B. (1996). The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, **6**(6), 329–348. https://doi.org/10.1016/0959-1524(96)00009-1

Wise, B. M., Gallagher, N. B., Butler, S. W., White, D. D., and Barna, G. G. (1999). A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, **13**(3–4), 379–396. https://doi.org/10.1002/(SICI)1099-128X(199905/08)13:3/4<379::AID-CEM556>3.0.CO;2-N

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. In *Multivariate Analysis* (1st ed., pp. 391–420). Academic Press.

Wold, S. (1978). Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models. *Technometrics*, **20**(4), 397–405. https://doi.org/10.1080/00401706.1978.10489693

Wold, S. (1992). Nonlinear partial least squares modelling II. Spline inner relation. *Chemometrics and Intelligent Laboratory Systems*, **14**(1–3), 71–84. https://doi.org/10.1016/0169-7439(92)80093-J

Wold, S., Esbensen, K., and Geladi, P. (1987a). Principal Components Analysis. *Chemometrics and Intelligent Laboratory Systems*, **2**(1–3), 37–52. https://doi.org/10.1016/0169-7439(87)80084-9

Wold, S., Geladi, P., Esbensen, K., and Öhman, J. (1987b). Multi-way principal components-and PLS-analysis. *Journal of Chemometrics*, **1**, 41–56. https://doi.org/10.1002/cem.1180010107

Wold, S., Kettaneh, N., Fridén, H., and Holmberg, A. (1998). Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemometrics and Intelligent Laboratory Systems*, **44**(1–2), 331–340. https://doi.org/10.1016/S0169-7439(98)00162-2

Wold, S., Kettaneh-Wold, N., MacGregor, J. F., and Dunn, K. G. (2009). Batch Process Modeling and MSPC. In *Comprehensive Chemometrics* (Vol. 2, pp. 163–197). https://doi.org/10.1016/B978-044452701-1.00108-3

Wold, S., Kettaneh-Wold, N., and Skagerberg, B. (1989). Nonlinear PLS modeling. *Chemometrics and Intelligent Laboratory Systems*, **7**(1–2), 53–65. https://doi.org/10.1016/0169-7439(89)80111-X

Wold, S., Sjöström, M., and Eriksson, L. (2001). PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, **58**(2), 109–130. https://doi.org/10.1016/S0169-7439(01)00155-1

Woodley, J. M. (2020). Advances in biological conversion technologies: New opportunities for reaction engineering. *Reaction Chemistry and Engineering*, **5**(4), 632–640. https://doi.org/10.1039/c9re00422j

Xu, L., Cai, C.-B., and Deng, D.-H. (2011). Multivariate quality control solved by one-class partial least squares regression: Identification of adulterated peanut oils by mid-infrared spectroscopy. *Journal of Chemometrics*, **25**(10), 568–574. https://doi.org/10.1002/cem.1402

Xu, L., Oja, E., and Suen, C. Y. (1992). Modified Hebbian Learning for Curve and Surface Fitting. *Neural Networks*, **5**(3), 441–457. https://doi.org/10.1016/0893-6080(92)90006-5

Xu, L., Yan, S. M., Cai, C. B., and Yu, X. P. (2013). One-class partial least squares (OCPLS) classifier. *Chemometrics and Intelligent Laboratory Systems*, **126**, 1–5. https://doi.org/10.1016/j.chemolab.2013.04.008

Yacoub, F., and MacGregor, J. F. (2004). Product optimization and control in the latent variable space of nonlinear PLS models. *Chemometrics and Intelligent Laboratory Systems*, **70**(1), 63–74. https://doi.org/10.1016/j.chemolab.2003.10.004

Yadav, A., Sharma, V., Tsai, M.-L., Chen, C.-W., Sun, P.-P., Nargotra, P., Wang, J.-X., and Dong, C.-D. (2023). Development of lignocellulosic biorefineries for the sustainable production of biofuels: Towards circular bioeconomy. *Bioresource Technology*, **381**, 129145. https://doi.org/10.1016/j.biortech.2023.129145

Yang, S., Navarathna, P., Ghosh, S., and Bequette, B. W. (2020). Hybrid Modeling in the Era of Smart Manufacturing. *Computers and Chemical Engineering*, **140**, 106874.

https://doi.org/10.1016/j.compchemeng.2020.106874

Yao, Y., and Gao, F. (2007). Batch process monitoring in score space of two-dimensional dynamic Principal Component Analysis (PCA). *Industrial and Engineering Chemistry Research*, **46**(24), 8033–8043. https://doi.org/10.1021/ie070579a

Yim, H., Haselbeck, R., Niu, W., Pujol-Baxley, C., Burgard, A., Boldt, J., Khandurina, J., Trawick, J. D., Osterhout, R. E., Stephen, R., Estadilla, J., Teisan, S., Schreyer, H. B., Andrae, S., Yang, T. H., Lee, S. Y., Burk, M. J., and Van Dien, S. (2011). Metabolic engineering of Escherichia coli for direct production of 1,4-butanediol. *Nature Chemical Biology*, **7**(7), 445–452. https://doi.org/10.1038/nchembio.580

Yin, S., Ding, S. X., Zhang, P., Hagahni, A., and Naik, A. (2011). Study on modifications of PLS approach for process monitoring. *IFAC Proceedings Volumes*, **44**(1), 12389–12394. https://doi.org/10.3182/20110828-6-IT-1002.02876

Yoon, S., and MacGregor, J. F. (2001). Incorporation of External Information into Multivariate PCA/PLS Models. *IFAC Proceedings Volumes*, **34**(27), 105–110. https://doi.org/10.1016/s1474-6670(17)33576-0

Yoon, S., and MacGregor, J. F. (2004). Principal-component analysis of multiscale data for process monitoring and fault diagnosis. *AIChE Journal*, **50**(11), 2891–2903. https://doi.org/10.1002/aic.10260

Zhang, Q., Li, P., Lang, X., and Miao, A. (2020). Improved dynamic kernel principal component analysis for fault detection. *Measurement*, **158**, 107738. https://doi.org/10.1016/j.measurement.2020.107738

Zhang, Y., and Qin, S. J. (2008). Improved Nonlinear Fault Detection Technique and Statistical Analysis. *AIChE Journal*, **54**(12), 3207–3220. https://doi.org/10.1002/aic.11617

Zhao, Z., Wang, P., Li, Q., and Liu, F. (2019). Product design for batch processes through total projection to latent structures. *Chemometrics and Intelligent Laboratory Systems*, **193**(May), 103808. https://doi.org/10.1016/j.chemolab.2019.07.007

Zhou, D., Li, G., and Qin, S. J. (2010). Total Projection to Latent Structures for Process Monitoring. *AIChE Journal*, **56**(1), 168–178. https://doi.org/10.1002/aic.11977

Zhou, H., Yu, K.-M., and Hsu, H.-P. (2021). Hybrid Modeling Method for Soft Sensing of Key Process Parameters in Chemical Industry. *Sensors and Materials*, **33**(8), 2789. https://doi.org/10.18494/SAM.2021.3436

Zhu, Q. (2021). Dynamic autoregressive partial least squares for supervised modeling. *IFAC-PapersOnLine*, **54**(7), 234–239. https://doi.org/10.1016/j.ifacol.2021.08.364

Zhu, Q., Liu, Q., and Qin, S. J. (2016). Concurrent Canonical Correlation Analysis Modeling for Quality-Relevant Monitoring. *IFAC-PapersOnLine*, **49**(7), 1044–1049. https://doi.org/10.1016/j.ifacol.2016.07.340

Zhu, Q., Zhao, Z., and Liu, F. (2021). Developing New Products with Kernel Partial Least Squares Model Inversion. *Computers and Chemical Engineering*, **155**, 107537.

https://doi.org/10.1016/j.compchemeng.2021.107537

Zhu, X., Rehman, K. U., Wang, B., and Shahzad, M. (2020). Modern Soft-Sensing Modeling Methods for Fermentation Processes. *Sensors*, **20**, 1771. https://doi.org/10.3390/s20061771

Zwick, W. R., and Velicer, W. F. (1986). Comparison of Five Rules for Determining the Number of Components to Retain. *Psychological Bulletin*, **99**(3), 432–442. https://doi.org/10.1037/0033-2909.99.3.432

Zydney, A. L. (2016). Continuous downstream processing for high value biological products: A Review: Continuous Downstream Processing. *Biotechnology and Bioengineering*, **113**(3), 465–475. https://doi.org/10.1002/bit.25695