UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Centro di Ateneo di Studi e Attività Spaziali "Giuseppe Colombo" - CISAS

CORSO DI DOTTORATO DI RICERCA IN SCIENZE, TECNOLOGIE E MISURE SPAZIALI
CURRICOLO: Scienze e Tecnologie per Applicazioni Satellitari e Aeronautiche
CICLO 32°

Design and Testing of Clustered Components For Modular Spacecraft Architectures

Tesi redatta con il contributo finanziario della fondazione cassa di risparmio di Padova e Rovigo

Coordinatore del Corso: Ch.mo Prof. Giampiero Naletto
Supervisore: Ch.mo Prof. Alessandro Francesconi

Dottorando : Francesco Feltrin

Abstract

In recent years, the space industry has demonstrated a renewed interest in multi agent systems, from the deployment of large and mega constellations to the plans to test Federated and in orbit assembly concepts. Furthermore, as the CubeSat platform has become a de-facto standard able to support ambitious missions, the cost for multi agent systems has decreased significantly.

The aim of this work is to study the benefits and drawbacks of large multi agent systems which might result from in-space assembly of numerous small autonomous spacecraft.

The thesis is divided into two parts; in the first we focus on how to efficiently and reliably control large clusters of actuators spread across a modular assembly. We examine both classical centralized and decentralized methods to solve the allocation of tasks within the clusters and finally propose a novel method, for which we provide proof of convergence and optimality. To characterize it, we simulate a large cluster of reaction wheels using data obtained from a hardware prototype. Compared to traditional methods, we observe reduced power consumption and more robust convergence when applied to large numbers of actuators. Finally, we generalize the model to encompass multiple inputs-multiple outputs systems. While multiple outputs can easily be accounted for, considering multiple inputs has revealed to be very challenging and only weak results are presented.

The second part is devoted to exploiting cluster properties during the preliminary design, leveraging both technological features and analytical conditions to improve design optimization methods. Building on the capabilities developed in the first part, namely the existence of an effective method to coordinate large number of actuators reliably, we present an analytical framework to pursue system design and optimization. A long and dry section of the thesis is devoted to the mathematical characterization of the framework and to provide proofs for its main properties. The abstract assumptions needed for the proposed algorithm are examined, and their validity assessed in the case of a CubeSat design procedure. Finally, a minimal computational implementation is described and applied to GOMX4-B mission.

Sommario

Il settore spaziale sta dimostrando un rinnovato interesse verso concetti basati sull'impiego di sistemi multi-agenti; dallo sviluppo di costellazioni con centinaia di satelliti (mega constellations) a test per architetture federate e dimostratori di assemblaggio in orbita. Inoltre, le piattaforme Cubesat sono ormai uno standard in grado di compiere missioni ambiziose, abbassando quindi il costo di sistemi multi agente. Questo lavoro si propone di studiare i benefici e gli svantaggi di sistemi composti da un grande numero di agenti, quali possono essere degli assemblati in orbita costituiti da innumerevoli satelliti autonomi.

Questo documento é diviso in due parti; nella prima ci si concentra su come controllare in maniera affidabile agglomerati composti da un grande numero di attuatori distributi su satelliti diversi. Vengono considerati sia algoritmi centralizzati che decentralizzati per risolvere il problema di allocazione dei compiti; viene infine proposto un nuovo metodo, per il quale vengono fornite dimostrazioni di convergenza. Per caratterizzarne il comportamento, si simula un cluster di ruote di reazione, modellate usando dati ottenuti con un prototipo da laboratorio. In confronto a metodi classici, l'algoritmo proposto mostra un consumo di potenza inferiore e una convergenza piú robusta soprattutto per grandi numeri di attuatori. Infine, si generalizza il modello di attuatore per comprendere anche casi con molteplici input e output. Mentre il caso di molteplici output viene trattato facilmente e differisce di poco dal caso con output singolo, trattare input multipli si é rivelato piuttosto complesso; vengono presentati solo risultati deboli.

La seconda parte é dedicata a sfruttare le proprietá dei cluster durante il design preliminare, facendo leva sia su caratteristiche tecnologiche che su proprietá formali per migliorare le procedure di ottimizzazione del desing. Mettendo a frutto i risultati ottenuti nella prima parte, ovvero la capacitá di coordinare in maniera efficace e affidabile un grande numero di attuatori, viene presentato un metodo analitico per l'ottimizzazione di sistema. Una lunga porzione della tesi viene dedicata a dimostrare le proprietá salienti del metodo. Le ipotesi necessarie per applicare il modello vengono esaminate per giudicarne la pretinenza nel caso di design di un cubesat. Infine, una implementazione computazionale viene descritta e applicata alla missione GOMX4-B.

Contents

Abstract **iii**

Sommario **v**

1 Introduction: Clusters, Redundancy and Multi Agent Concepts **1**
1.1 Motivation 1
1.1.1 The fractionated concept 1
1.1.2 Beyond fractionated 3
1.2 Premise and structure of the thesis 4
1.2.1 Structure of the work 5

2 Decentralized Cluster Control **11**
2.1 Problem statement and general idea 11
2.2 Literature review 12
2.2.1 Proposed algorithm and qualitative comparison 15
2.3 Formal model definition 16
2.3.1 Individual agent 16
2.3.2 Cluster 17
2.3.3 Dual ascent 18
2.3.4 Method of multipliers 19
2.4 Formal results statement 20
2.5 Proofs 21
2.5.1 Proof of convergence to stationary points along the constraints 22
2.5.2 Proof of local stability of constrained minima 24
2.5.3 Instability of maxima and saddle points 25
2.5.4 Speed characterization 26

3 Application to small sat ADCS **31**
3.1 Analytical model of a RW 31
3.1.1 Power consumption model 32
3.2 Hardware characterization 33
3.2.1 Brief description of the hardware 34
3.2.2 Open loop steady state characterization 35
3.2.3 Open loop dynamic characterization 38
3.2.4 Close loop control 41

3.2.5 Notes on real time behavior 46
3.3 Cost functions 47
3.3.1 Statistical approach 49
3.4 Allocation strategies comparison 52
3.4.1 Preliminary results with pairs of agents 53
3.4.2 Static allocation 54
3.4.3 Distributed allocation 57
3.4.4 Efficiency trends with larger numbers of agents 58
3.4.5 Robustness under discrete time implementation 60
3.4.6 Effect of different λ_0 63
3.4.7 Effect of random noise 64
3.4.8 Dynamic request 65

4 Generalized Clusters 71
4.1 Extending to MIMO 71
4.1.1 Multiple input generalization 71
4.1.2 Multiple outputs for each agents 72
4.2 An endogenous cost function 77
4.2.1 Model notation and hypothesis 80
4.2.2 Example with positive monomial **E** 81
4.2.3 Case for general **E** 84
4.2.4 Design of the cost function 87

5 Analytical System Design Using Cluster Scale Laws 91
5.1 Literature review on Multi Disciplinary Optimization 91
5.2 Formal general idea 95
5.2.1 The mission worthy set $\mathcal{A}$ 98
5.2.2 The technologically worthy set $\mathcal{B}$ 99
5.2.3 Minimization of cost function 101
5.3 Existence of a strong optimum 102
5.3.1 Model set up 107
5.3.2 An example toy problem 108
5.3.3 Proof of $\mathcal{F}$ contraction map 109
5.3.4 Practical methods to prove existence of $\vec{c}$ 113
5.4 Consequences of Lemma 10 115
5.4.1 External fluxes 115
5.4.2 Modeling primary storage 115
5.4.3 Secondary capacity 116
5.5 Non ideal systems central Lemma 118
5.6 Proofs 122
5.6.1 Lemmas used in the second model 127

6 A CubeSat Application 133
6.1 Model for ACS 135
6.2 Model for EPS 135
6.3 Model for Thermal Control 140
6.4 Model for On Board Computer 141

6.5 Model for Telecommunication . . . 142
6.6 Model for Propulsion . . . 143

7 An Example Implementation 145
7.1 Overview of the software . . . 145
7.1.1 Mission definition . . . 146
7.1.2 Subsystem design . . . 148
7.2 Test mission and results . . . 149
7.2.1 GOMX-4B . . . 150
7.2.2 4B-Mission implementation . . . 151
7.2.3 Results . . . 155

8 Conclusions 159

List of Figures 165

List of Tables 167

Chapter 1

Introduction: Clusters, Redundancy and Multi Agent Concepts

1.1 Motivation

Clusters *per se* are nothing new.
They are often found as patterns of repeated components used to increase the reliability of a critical section of a system. Both passive (Triple Modular Redundancy) and active architectures (stand-by sparing and pair-and-spare) have been used extensively from the early stages of the space industry. In the Gemini capsule, the tapes used by the on board computer had each bit repeated three times and a voter circuit was applied to read the information. In the Saturn 5, the logic section of the launch vehicle digital computer (LVDC) used TMR at the circuit level [1]. In the Space Shuttle orbiter, four computers were used in a redundant set for the most critical phases [2] while keeping a fifth one for backup[1]; four redundant actuators drove the hydraulics of each of the aerodynamic surfaces and two computers were used for each of the three main engines. Finally, each of the nine engines in a Falcon 9 rocket is controlled by the voting of three computers and the system can lose up to two of motors and still complete the mission.

However, recent and bolder mission concepts enable a different kind of clusters. These new, virtual, agglomerates can emerge from a temporary collaboration between different spacecraft. As such, they can be used in more complex functions but they also need to respond to more challenging requirements, such as working within dynamic architectures where detaching or adding modules might be very frequent. We begin with an overview of the path that led to the new distributed architectures for space missions.

1.1.1 The fractionated concept

With the merely technical exception of constellations, the monolithic architecture has been the go-to solution of the space industry and, in retrospect, it has been very successful. This is not

[1] In fact, Arnold Aldrich, Director of the Shuttle Office at Johnson Space Center argued for an additional computer to be carried along as a spare and indeed, the sixth computer was carried on the first few flights.

to say that it has no drawbacks and, in the mid 2000s, the F6 project (Future, Fast, Fractionated, Free-flying Spacecraft united by Information exchange) was put forward to address them.

Large monolithic spacecraft are typically custom made and tailored to an individual mission. The high level of optimization that this allows comes at the expenses of lengthy design phases, even longer test and integration campaigns, and limited possibilities for future re-use of the same solutions. Overall, this has contributed to poor standardization, which in turn exacerbated premature aging of the system, where to remedy the obsolescence of a few key components, a long and expensive redesign on system level is required. Moreover, by the beginning of the 2000's it had become painstakingly clear that large projects are intrinsically complex to execute and prone to be over budget and behind schedule[3],[4],[5].

The solution proposed by Brown and Eremenko in a series of papers [3],[4],[5] was the fractionated architecture, a system composed of multiple independent and free-floating agents which were to behave as a single system. The concept was to be developed during the F6 project, funded by DARPA with the goal of flying a demonstrator by 2015[10].

The core idea to solve the problems of large projects was to break them down into independent modules which could be developed and deployed more independently. Besides enabling the use of more diverse launcher (possibly opting for smaller and cheaper alternatives) and allowing bigger infrastructure to be deployed, such as large telecopes, the concept was proposed as a way to address economic and business oriented concerns. The transition from the exploratory phase of a project to the more mature and profitable one would be smoother, and the staged deployment of assets could be used to test the market and reduce the risks of catastrophically poor economic predictions, as in the case of the Iridium constellation. Notably, the strategy of staged constellation deployment[6], is currently used by the Starlink project from Space-X.
The fractionated concept was also seen as a way to enhance traditional risk mitigation, both in the event of launcher and system failure. In the event of a launcher accident, the loss would impact only part of the program, slowing down deployment or augmentation but not stopping it completely. Also maintenance through the replacement of faulty units would be easier, leveraging the concept of graceful degradation. Even though an occasional fault would reduce the capabilities of the system, most functionalities would be still available and fixing the issue would not have been as expensive as in the traditional case. Another key goal of the program was the advancement of open interface standards, to facilitate the re-use of components, to speed up the design phase, and to allow multiple partners to work on the same project independently.
On a side note, military-friendly capabilities were also part of the program, especially the ability to perform a defensive scattering maneuver to avoid *debris-like* threats and return to formation without ground intervention.

At the sytem level, several other potential benefits were ascribed to the physical decoupling of the modules. Distributed versions of the classical subsystem were proposed, pursuing both subsystem specific advantages as well as a general improvements due to the disassociation of different disciplines and their different requirements. For example, some modules could have been devoted to power production and/or storing, therefore freeing payload-specialized modules from having to point their solar panel toward the sun. Short distance infra-module communication would have enabled distributed computations and data storage while communication with earth would have been routed through only a few, specialized long range radios. Daring concept for distributed thermal management were proposed where the target would be illuminated by either concentrated sunlight [7], laser [8] or microwave [9] and cooled by having an active module shielding the target one from the sun. Finally, for the basic functions which could not be effectively distributed, identical subsystems would be included in each module to promote standardization and leverage economies of scale.

Despite the multitude of arguments in favor of the fractionated architecture, the F6 project was canceled in 2013 and the idea never found a faithful implementation. This is certainly due to a variety of concurrent reasons, however, we point to three of technical significance

- The technology for efficient wireless power transfer was not, and still is not, mature.
  Different technologies have been proposed depending on the distance between the generating and receiving satellite. For near field transfer (less than a meter), electrodynamic induction can be quite efficient (up to 95%) but imposes extremely demanding requirements on orbital control; efficiency rapidly drops at larger distances [14],[15]. For long range power transmission, in the order of km, both microwave and laser systems have been considered in areospace applications [16]. They operate with the same philosophy [17], the first using a microwave emitter and a rectenna or hemispherical antenna [19], while the second using a solid state laser and an optimized photo voltaic panel [20]. Both implementation suffer from similar drawbacks; very low end-to-end efficiencies (less than 45% for microwave [21] and possibly less than 25% for laser [18]) and demanding requirements on the ADCS of both satellites.

- Launcher vehicles have become more reliable [11], so the insurance effect granted by spreading the system deployment across multiple launches has less of an impact on the predicted life cycle cost of the mission compared to expectations obtained when the concept was first proposed [12], [13].

- The rapid development of the cubesat standard has met many of the core concerns that the fractionated concept set out to address, namely to reduce development time, cost and enable standardization.

1.1.2 Beyond fractionated

Despite the premature ending of the F6 project, milder version of the fractionated spacecraft have been proposed and are scheduled for, or have already been flown on, cubesat demonstrator missions. They still feature multiple spacecraft (although typically just two due to the proof-of-concept nature of these missions) which bypass the wireless power transfer problem either by not transmitting power, as in the case of the federated spacecraft, or by docking together mechanically.

The concept of Federated satellites [27],[28],[29] is to have satellites from different owners collaborate on an opportunistic basis; when an under-utilized resource is available on a satellite, as for example memory or processing power, it can be rented out to another spacecraft. Cubesat demonstrators have been flown (GOMX4A/B) and are scheduled to fly sometime between late 2019 and early 2020 (FSSCat). The most immediate and easily implemented behavior is the extension of downlink capabilities between multiple spacecrafts, but sharing of memory and computational capabilities might also be achieved.

If more capabilities are to be shared, physically attaching the various spacecraft is an attractive possibility [34], [35],[34]. Problems regarding wireless transmission are eliminated, although some of the initial advantages of F6 are also lost in the trade. Multiple programs to demonstrate in orbit assembly on cubesat platforms are on their way (C-Pod, RACE). This can be done both for short and medium term collaborations, like re-supply and on orbit servicing missions [30],[31],[32].

1.2 Premise and structure of the thesis

Cumulatively, we can view fractionated, federated and in-orbit assemblies as *multi agent* concepts. Contrary to monolithic architectures, which use redundancy to improve reliability,in multi agent systems, redundancy emerges organically from the duplication of modules and capabilities, as each agent needs to be able to operate autonomously to some extent. Then, we can envision a *virtual* cluster, made out of components spread out across different spacecraft which happen to be involved in a temporary collaboration.

As a consequence, the functions endowed with redundant actuators are extended well beyond the safety critical ones; potentially all tasks could be met by a coordination of different agents. This abundance of degrees of freedom enables the pursuit of secondary objectives; they could be exploited to optimize some performance (such as efficiency), increase reliability or, as we will show shortly, to improve both. However, there are many potential pitfalls which need to be carefully considered and, if possible, addressed during the design of the control mechanism. The most glaring example is the increasing computational cost required for the coordination of large number of agents; depending on the chosen protocol, the number of messages exchanged between the nodes might increase linearly or quadratically. This cost alone could be enough to render large agglomerate unreasonably ineffective. Furthermore, the temporary nature of this assemblies injects significant complications into the problem. We need to consider the agglomerate behavior under the assumption of a dynamic architecture. A spacecraft/node might unexpectedly become unavailable, either due to a malfunction or its necessity to leave the network; others might join the agglomerate, possibly without disrupting ongoing activities.

The wish to combine multiple agents to enhance their capabilities should also consider the possibility of heterogeneous components, as originally envisioned in the F6 program. Even with similar spacecraft the individual components might leverage different technologies for the same task. As newly deployed systems are introduced to replace older ones, a complex blend of diverse actuators need to be controlled. To enable an inexpensive integration of different components, the algorithm chosen to coordinate the cluster will need to exploit some common interface, rather than a more direct protocol.

Finally, as is recently becoming necessary for distributed control algorithms, some consideration should be given to the possibility of malicious agents. Having many nodes, possibly owned by different entities, interacting in uncontrolled ways might introduce reliability concerns beyond simple failure or glitches. Preventing the spreading of generally undesired behaviors should therefore not only consider random failures which might trigger chain failures, but also the possibilities of intentionally disruptive actions.

We have provided a general introduction to the challenges that we need to address in order to seriously consider the deployment of large multi agent system. On the other hand, the benefits could be substantial. The aim of this work is to study the challenges of a cluster architecture, and its potential.

1.2.1 Structure of the work

We will look at clusters through two lenses; how to control and coordinate them effectively and how to design for/with clusters.

The first addresses many of the concerns raised in this introduction and offers a method to reconcile reliability and efficiency. It is heavily based on distributed and decentralized control techniques. Initially, the essential mathematical model is presented, based on single input-single output actuators in a very general and abstract way. Then a concrete implementation of the method is discussed an applied to ADCS for small satellites with numerical experiments. Finally the model is expanded to a Multiple inputs-Multiple outputs framework, but only partial results have been found.

The second part aims to exploit the cluster properties to improve the overall system design phase. Here we explore the simple and appealing scaling properties of clusters. We consider the problems of Multidisciplinary Design Optimization (MDO) and discuss how clusters might mitigate them. Then we present a tailored approach for clusters which has significant analytical benefits compared with MDO. In the following chapter we discuss the model giving emphasis to its application to CubeSat design and finally provide a computational implementation.

A chapter-by-chapter break down of the work follows:

1. Decentralized cluster control
   The aim of this section is to present the novel control algorithm developed. The requirement for the control scheme is to solve the constrained optimization of a generic objective function while preserving the high reliability typical of redundant architectures. The discussion is framed in formal and general mathematical terms to facilitate the comparison with the literature on constrained optimization. Classical centralized and decentralized method to solve the problem are examined critically and found ill suited for the application. The proposed method is introduced along with formal claims about convergence to the constrained local optima and proofs.

2. Application to small sat ADCS
   To characterize the algorithms proposed in the previous section, we apply it to a cluster of Reaction Wheels (RW), which could be used to control the attitude of a small satellite. After presenting an analytical model for the RW, an hardware demonstrator is used to validate it and to fit its specific parameters. Then, the numerical model for the RW is used to simulate the behavior of a cluster with a large number of actuators. Characterization of the proposed algorithm is pursued to validate the decrease in power consumption with respect to traditional methods of coordination and to demonstrate convergence in a discrete time implementation.

3. Generalized clusters
   This final section on control generalizes the model used for the individual actuator, from single input- single output to multiple inputs - multiple outputs. In the first part, multiple outputs are considered. The mathematical model is presented and, following the same structure used in the single output case, analytical proofs are provided. The second part revolves around the use of a cost function to assign a single value to the vector of consumed resources, the multiple inputs. A cost function internal to the system is proposed but only weak results have been found to support its merit.

4. Analytical system design using cluster scale laws
   This chapter marks the beginning of the second part of the thesis, where the focus shifts from controlling a cluster already present in a system to designing a new system which features clusters. We start with a review of the state of the art in Multidisciplinary Design Optimization (MDO), a discipline aimed at studying the effectiveness of different computational architectures for large constrained optimization problems. The drawbacks of the MDO techniques are used as a motivation to develop a new design framework. The benefits of using clusters within a standard MDO are briefly discussed before developing the more tailored approach. A formal model for system design is presented along with the major result; namely the existence and uniqueness of a design which is optimal for a large class of cost functions. The remainder of the chapter is devoted to exploring how to verify the hypothesis needed for the main result.

5. A Cubesat application
   Using the analytical results showed in the previous chapter, we apply them to a cubesat system. We review the major subsystems and the formulation we could use to model them while abiding the hypothesis needed by the analytical framework.

6. An example implementation
   Finally, to test the applicability of the analytic framework developed in the previous chapters, we implement it numerically and apply it to realistic mission scenarios. In particular, we focus on small satellites used for earth observation missions.

Bibliography

[1] IBM 1964. "Saturn V Launch Vehicle Digital Computer, Volume One: General Description and Theory"

[2] IBM, J. R. Sklaroff (1976). "Redundancy Management Technique for Space Shuttle Computers".

[3] Brown, Owen; Eremenko, Paul (2006). "The Value Proposition for Fractionated Space Architectures" . AIAA Space 2006. San Jose, CA: American Institute of Aeronautics & Astronautics. pp. Paper No. AIAA 2006 7506

[4] Brown, Owen; Eremenko, Paul (2006). "Fractionated Space Architectures: A Vision for Responsive Space" . 4th Responsive Space Conference. Los Angeles, CA: American Institute of Aeronautics & Astronautics. pp. Paper No. AIAA RS4 2006 1002.

[5] Brown, O.; Eremenko, P.; Roberts, C. (2006). "Cost-Benefit Analysis of a Notional Fractionated SATCOM Architecture", 24th International Communications Satellite Systems Conference. San Diego, CA: American Institute of Aeronautics & Astronautics. pp. Paper No. AIAA 2006 5328

[6] O. deWeck, R. deNeufville, and M. Chaize (2003). "Enhancing the economics of communications Satellites via Orbital Reconfigurations and Staged Deployment". In AIAA Space 2003 Conference.

[7] R. R. Secunde and T. L. Labus. Solar Dynamic Power Module Design. In 24th Intersociety Energy Conversion Engineering Conference, 1989.

[8] P. E. Glaser. Power from the Sun: Its Future. 162(November):857-861, 1968. Science Magazine, 162(November):857-861, 1968.

[9] R. H. Dietz. Solar Power Satellite Microwave Transmission and Reception. Technical report, NASA, 1980.

[10] Broad Agency Announcement, System F6, Tactical Technology Office (TTO). Technical report, DARPA, 2010. DARPA-BAA-11-01.

[11] SpaceNews. Space Launch Report https://www.spacelaunchreport.com/log2018.html accessed August 2019.

[12] D. R. Sauvageau and B. D. Allen (1998). "Launch Vehicle Historical Reliability". In 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit

[13] S. D. Guikemaa and M. E. Paté -Cornell (2005). "Probability of infancy problems for space launch vehicles" . Reliability Engineering and System Safety, (87):303-314.

[14] M. J. Simon, C. Langer, S. Rubin, D. Komush, and D. B. Maciuca (2009). "Wireless Power Transfer for Responsive Space Applications". In AIAA SPACE 2009 Conference & Exposition.

[15] D. J. Barker and L. Summerer (2011). "Analysis of near-field wireless power transmission for fractionated spacecraft applications". In 62nd International Astronautical Congress.

[16] T.J. Nugent and J. T. Kare. Laser Power for UAVs. Technical report, LaserMotive, 2010.

[17] Nayfeh, Fast, Raible, Dinca, Tollis, and Jalics (2011). "High Intensity Laser Power Beaming Architecture for Space and Terrestrial Missions". Technical report, NASA.

[18] T. Nayfeh, B. Fast, D. Raible, D. Dinca, N. Tollis, and A. Jalics (2011). "High Intensity Laser Power Beaming Architecture for Space and Terrestrial Missions". Technical Report NASA/TM-2011-217009, NASA.

[19] C. T. Rodenbeck, M. yi Li, and K. Chang (2004). "A Phased-Array Architecture for Retrodirective Microwave Power Transmission from the Space Solar Power Satellite". In Microwave Symposium Digest, 2004 IEEE MTT-S International

[20] C. T. Bellows, N. M. Keller, and J. T. Black (2010). "Mission Feasibility Study for Space Based Wireless Power Transfer". In AIAA/AAS Astrodynamic Specialist Conference

[21] R. M. Dickinson (2003). Wireless Power Transmission Technology State of the Art. Acta Astronautica, 53:561-570

[22] Jing Chu, Jian Guo, Eberhard Gill (2016). "Decentralized autonomous planning of cluster reconfiguration for fractionated spacecraft". Acta Astronautica 123 397-408

[23] Mohsen Mosleh, Kia Dalili, Babak Heydari (2014). "Optimal Modularity for Fractionated Spacecraft: The Case of System F6". Conference on Systems Engineering Research.

[24] Gregory F. Dubos n , Joseph H. Saleh (2011). "Comparative cost and utility analysis of monolith and fractionated spacecraft using failure and replacement Markov models ". Acta Astronautica 68, 172-184

[25] YAO Wen, CHEN Xiaoqian , ZHAO Yong, Michel van Tooren (2012). "A Fractionated Spacecraft System Assessment Tool Based on Lifecycle Simulation Under Uncertainty". Chinese Journal of Aeronautics 25, 71-82

[26] Tatiana Kichkaylo, Lucy Hoag, Elizabeth Lennon, Gordon Roesler. (2012). "Highly Efficient Exploration of Large Design Spaces: Fractionated Satellites as an Example of Adaptable Systems". Procedia Computer Science 8 428-436.

[27] Alessandro Golkar, Ignasi Lluch Cruz (2015). "The Federated Satellite Systems paradigm: Concept and business case evaluation". Acta Astronautica 111 230-248

[28] Rustam Akhtyamov, Ignasi Lluch Cruz (2016) "An implementation of Software Defined Radios for federated aerospace networks: Informing satellite implementations using an inter-balloon communications experiment". Acta Astronautica 123, 470-478.

[29] Olga von Maurich, Alessandro Golkar (2018). "Data authentication, integrity and confidentiality mechanisms for federated satellite systems". Acta Astronautica 149, 61-76.

[30] A. Long (2005). "Framework for Evaluating Customer Value and the Feasibility of Servicing Architectures for On-Orbit Satellite Servicing". Master's thesis, Department of Aeronautics and Astronautics and Engineering Systems Division, Massachusetts Institute of Technology.

[31] C. Reynerson (1999). "Spacecraft Modular Architecture Design for On-Orbit Servicing". AIAA, 99(4473).

[32] C. Joppin and D. Hastings (2006)." On-Orbit Upgrade and Repair: The Hubble Space Telescope Example". Journal of Spacecraft and Rockets, 43(3):614-625.

[33] Erica L. Gralla, Oliver L. De Weck (2006). "Strategies for on-orbit assembly of modular spacecraft". JBIS, Vol. of 60, pp.219-227,

[34] Craig Underwood Sergio Pellegrino, Vaios J. Lappas, Christopher P. Bridges, John Baker (2015). "Using CubeSat/micro-satellite technology to demonstrate the Autonomous Assembly of a Reconfigurable Space Telescope (AAReST)". Acta Astronautica 114, 112-122

[35] Lorenzo Olivieri, Alessandro Francesconi (2016). "Design and test of a semiandrogynous docking mechanism for small satellites". Acta Astronautica 122, 219-230.

[36] Daniel Selva , David Krejci (2012). "A survey and assessment of the capabilities of Cubesats for Earth observation". Acta Astronautica 74, 50-68.

[37] Armen Poghosyan , Alessandro Golkar (2017). "CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions". Progress in Aerospace Sciences, Volume 88, Pages 59-83

Chapter 2

Decentralized Cluster Control

2.1 Problem statement and general idea

The allocation problem Many tasks in engineering can naturally be performed by multiple agents working in parallel. Parallelization can increase throughput, efficiency, performance and improve system reliability. For example, modern computer architectures feature multiple cores and multiple CPUs in parallel to increase throughput. In the automotive field, commercial hybrid vehicles as well as high end sport cars use auxiliary electric motor(s) to augment the main internal combustion engine. The strategy to allocate power split among the engines is chosen to either maximize efficiency (in the first case) or some performance, like acceleration (in the second). In commercial passenger flight, strict regulations require that no single failure would jeopardize the safe operation of the airplane. Hence, modern airliners feature multiple engines that work in parallel to supply the thrust needed to maintain speed and to power the electrical system of the aircraft. As it happens, in-flight shutdown are fairly common and, thanks to the redundancy of the propulsion system, harmless.

In general, if a process can be decomposed into independent tasks, not constrained by a specific order of execution, multiple agents can simultaneously contribute to the completion of the process. We have freedom in the choice of the number of agents to assign to the process as well as the pairing of tasks and agents. Among the combinations of i) process division into tasks and ii) agent-to-task matching that are able to complete the process (i.e. the feasible combinations), it is natural to select the one that minimizes some cost function pertinent to the problem considered.

In this chapter, we will consider the number of agents to be fixed by the design of the system and we will focus on how to optimally allocate the tasks among the agents. To make our objective more concrete, we can consider the example of the airliner; the process is the production of a requested thrust, the sub tasks are the different possible way of producing this cumulative output, the cost function is total fuel consumption and a qualitative strategy to minimize it might be to allocate fractions of the requested thrust to each agent according to its efficiency, where the most efficient engine is tasked to produce the largest share of total thrust.

2.2 Literature review

The problem can be rightfully framed as a constrained optimization problem: minimize a cost function (for example energy consumption) subject to the exact constraint of accomplishing the task (to collectively produce the requested power). In this section, we present a high level overview of the macro categories available to solve this problem, highlighting the typical advantages and drawbacks of each. The first macro division is between centralized and decentralized methods, with the information path depicted in Fig 2.1. Among decentralized algorithms, a further differentiation will be highlighted, by either assuming collaborative or non-cooperative agents.

Centralized approach The Lagrange multipliers method is a classical example of centralized algorithm; a single entity, having the full picture of the network state, computes the optimal solution and instructs all other agents on what to do. Calling $X \in \mathbb{R}^n$ the vector of allocation, which represent the quantity of output each agent must produce, $G(X)$ the function of cumulative consumption to minimize and $S(X) = \vec{0}$ the set of constrains, the stationary points for the constrained optimization are found by solving for the stationary points of $\mathcal{L}(X, \vec{\lambda})$.

$$\mathcal{L}(X, \vec{\lambda}) = G(X) + \vec{\lambda} \cdot S(X) \qquad \nabla \mathcal{L}(X_0, \vec{\lambda}_0) = \vec{0} \tag{2.1}$$

This method requires all information to be considered at the same time. The central node needs to know in which way each agent contributes to the cumulative consumption $G(\cdot)$ and solve a typically non linear problem. Moreover, in typical engineering applications, one can not assume that the model for $G(\cdot)$ is time independent or that it can be fixed during the design phase. Although this would spare the central node from having to poll data from the whole network, any failure in a component must be accounted for by the model used to solve the problem. Not updating the model would lead to sub-optimal solutions (in the best case) or constraint violation (in the worst). In practice, some form of handshaking is required between the network and the central node. Implementation might vary from low bandwidth protocols used only to monitor the status of the agent (working/failed) to more sophisticated communications used to track slow drifts in performances and adjust the optimal allocation accordingly.

The centralized approach, although conceptually very simple, is plagued by communication and computational bottle necks and single points of failure. In general, as the central node has to handle all commands and updates within the network, the maximum number of agents is bounded by the capabilities of the central node; the architecture does not scale well with the number of agents.

In response to these drawbacks, considerable effort has been devoted to the development of a decentralized version of Lagrange methods. Notable examples are the Dual Ascent method, the Method of Multipliers (MM) and the Alternating Direction Method of Multipliers (ADMM). Dual Ascent (DA) requires convexity of the cost function, which as we will see in the next chapter, is not always possible to guarantee. ADMM solves this problem by introducing a modified Lagrangian to tolerate non convex functions. These methods are promising for our applications and will be explored in more details in sections 2.3.3 and 2.3.4.

Decentralized algorithms The desire to solve a problem in a decentralized way ensues naturally in a vast number of practical situations. It may be suggested by the nature of the system implementation, like in a sensor network or in a flock of drones, where agents are spread over a large region

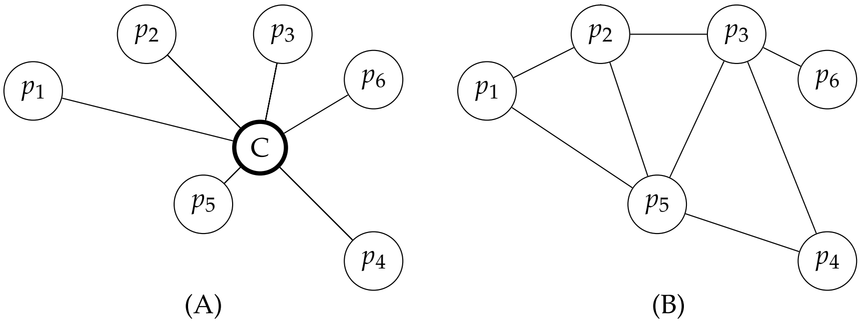


Figure 2.1: Information paths in Centralized (A) and Decentralized (B) algorithms. In the centralized approach, the node C has to process all information about the network. With the distributed approach, each agent handles only a subset of the total information handling.

of space and communication capabilities might be at a premium. Various methods to ensure the achievement of a common goal through the individual actions of the agents have been proposed in recent years; a much more detailed review can be found in books on multi agent systems, such as [1]. To provide an overview of the variety of possible approaches, we differentiate using the assumption at the base of the agent's behavior; whether it is collaborative or non cooperative.

In collaborative settings, imperative instructions are duly performed by every agent; the approach is algorithmic. Computer science and distributed control are fields that subscribe to this assumption. In non-cooperative settings, agents are free to choose how to behave in order to maximize some personal goal (such as profit maximization). This apparently unnecessary complication is useful to naturally handle local constraints: agents choose their best option among a set of feasible actions, which depends on local information. Game theory, and in particular the idea of Nash equilibrium, is the framework used to study the agents limit behavior[1]. In economics, mechanism design is the tailoring of the specific game rules that favors the emergence of a desired collective behavior.

Collaborative decentralized algorithms Collaborative (or deterministic) agents can coordinate by either peer-to-peer communication within a neighborhood or by interacting in an one-to-many, aggregated way. Algorithms that employ the first mechanism are called distributed optimization algorithm while the second are known as contract nets.

Distributed optimization allows n agents to coordinate and minimize the sum of convex functions $f(x) = \sum_{i=1}^{n} f_i(x)$ subject to a global state x. Each function $f_i(\cdot)$ is known only by agent i. Intuitively, to minimize f all agents need to first agree on the same global state x. A well studied framework to achieve this is called *consensus*[9][11], which refers to a group behavior designed to ensures that all agents asymptotically reach a common estimation of the global state. For example, assuming x to be a scalar for simplicity and $\hat{x} \in \mathbb{R}^n$ to be the vector of the state estimates by each of the n agents, a consensus algorithm can guarantee that eventually $x_i = x_j$ for any two agents

[1] Assuming some learning on behalf of the agents, eventually they will reach some Nash equilibrium, at least for the subset of games we are considering.

i, j. A typical implementation is

$$\hat{x}(t_{i+1}) = A \cdot \hat{x}(t_i) \qquad A \in \mathbb{R}^{n \times n} \tag{2.2}$$

Where A is a stochastic matrix[2]. An intuitive interpretation of the law 2.2 is that agent's i next state, $x_i(t_{i+1})$, is some weighted average of the states of its neighbors. Conditions for consensus are presented in many papers such as [5], [6] or [7],[8] in more general, time varying graph topologies. The decentralized gradient[11],[9],[10] or sub-gradient[12] algorithms used in distributed optimization are based on some version of consensus.

To apply distributed optimization to the allocation problem, one needs to find a way to enforce constraint satisfaction, for example by incorporating the constraint in the global state. More sophisticated solution implement constraints with projection techniques[13],[14], where equality constraints are assumed to be time invariant and linear with the state. For a more detailed review of distributed coordination, we refer to [15].

The drawbacks of using a consensus based approach is that performace depends on how connected the graph is; to converge more quickly, a higher level of interconnection is needed, which require more messages to be exchanged. More importantly, as the communication graph underling the network must be at least connected[3], a malicious or faulty agent could hijack the whole network; either preventing convergence or forcing it towards a wrong value.

Contract nets[2],[3],[4] are a framework/protocol to distribute a set of tasks to a set of agents through some form of negotiation. Each agent may have different capabilities and therefore incur in a different cost to accomplish a specific task. Given a random initial pairing task-agent, it will likely be sub-optimal, meaning that the sum of all costs that the agents would have to sustain to accomplish the assigned tasks is not minimal. The agents thus enter a negotiation process where they exchange tasks and (virtual) currency in order to reduce their cost. Under appropriate circumstances, it can be proved that the minimization of individual cost leads to a optimal task assignment. Once the allocation phase is over, each agent executes its respective task.

The choice of the negotiation process defines the properties of the contract net. A popular choice is an auction where, for each task to be assigned, each agent bids its marginal cost[3], which is the agents additional cost caused by the addition of the new task to its current workload. The task is assigned to the lowest bidder. Then the process repeats. Note that this process is communication intensive as all agents that are capable to bids on a task do so. Also, to use this method in a constrained optimization setting, the set of tasks should be chosen as to satisfy the constraint by design.

Non-cooperative/self interested agents In non-cooperative settings, self interested agents interact using a predefined mechanism (an auction, a market place etc). As we can not force agents to behave in arbitrary ways, the interaction scheme (the *game*) is designed to promote desired behaviors. Such emerging behaviors can be found by studying the Nash equilibria of the specific game.

Similarly to contract nets, consider an auction in which a task is to be assigned to one of the multiple agents who bid on it. Each agents privately communicates to the auctioneer the (minimum) price at which it would be willing to perform the task. In this setting, the agents are not

[2] Each element is non negative and the row sum is equal to 1

[3] Here we mean that there exists at least one agent that has a sequence of directed edges that connects it to every other agent in the network. Under this assumption the directed graph as a directed spanning tree which is the weakest condition needed for consensus in time varying graphs[7][8].

forced to bid their true cost for the task (such as their marginal cost); instead they will try to maximize their expected utility regardless of the impact that their behavior might have on overall network. However, if we implement a second-price sealed-bid auction[16][4], in which the winner (lowest bidder) is awarded with the second lowest price, it is possible to prove that expected utility is maximized by bidding the true valuation/cost[5]. Therefore, with an appropriate choice of mechanism design, we can expect self interested but rational agents to effectively collaborate.

In practice, the local objective that each agents tries to maximize is a degree of freedom for the control engineer; we are not limited to the expected utility. Under certain assumptions for the game[6], it is possible to determine a set of local objective functions which is aligned with a given global objective function.

The implementation for control application [17][18][19] follows two steps;1) defining the agents local objective from the desired collective behavior and 2) specifying the learning algorithm that allows each agent to improve their strategy choice and reach the desired Nash equilibrium. This hierarchical decomposition, where we can separate the design of the global behavior from that of the agent is the strength of this approach. Contrary to the deterministic counterpart, we do not need to specify the response of each agent to every possible environmental condition; we only assume that they will maximize their preference among the feasible choices they have, hence automatically considering local constraints.

2.2.1 Proposed algorithm and qualitative comparison

Consider n agents collectively required to produce a specific output while minimizing resource consumption. We propose that each agent chooses how much to contribute only on the basis of an aggregated variable, which can be accessed either by direct measure or is distributed by a public broadcast. In the latter case, we can imagine a separate entity outside the cluster, an *announcer*, that broadcasts the value of the variable without any knowledge about the network or any direct acknowledgment from it. To motivate why we might want to subscribe to this scheme, we will compare it to the literature presented above. It is worth noticing that the proposed algorithm is designed *ad hoc* for the parallel production problem; generality is traded for greater reliability, scalability and ease of implementation.

The implementation is decentralized by design, which eliminates bottlenecks in communication and computation as well as possible single points of failure. Every agent is able to choose its next output without having to communicate with any other, so communication effort does not increase with the number of agents. Even when using an external announcer, the value to be transmitted does not depend on the number of agents, which indeed is unknown, hence the computational cost to produce this value is fixed with n. Finally, while the announcer may appear as a single point of failure, it is not necessarily so. There is no need for it to be unique; multiple announcers could be used in a majority vote[7] scheme.

When compared with distributed optimization, by forbidding agent to agent communication, the system is more

- Robust, as it can not be hijacked as consensus can; moreover agents can connect and disconnect at will without disrupting the network operations

[4] Also known as Vickrey auction or Vickrey-Clarke-Grove Auction in the most general form.

[5] This is a well known result in game theory: in [1] the proof can be found as Theorem 11.1.1, pp. 333

[6] Such as for *potential games* or *weakly acyclic games*

[7] Sometime this architecture is referred as a k-out-of-n system

- Scalable; as the number of agents is not known by the architecture, it virtually has no effect on it. In practice, if an announcer is used, the strength of the signal may need to be adjusted for the number of agents.

Contract nets (and auctions in general) show many similarities with the proposed scheme, as in both settings the agents react to an common signal. For example, in an English auction, one can view the auctioneer in the role of the announcer. As bids are made, the price increases and is communicated to all participants, just as the global variable is broadcasted to all agents in the network. However, the auction requires a communication from the agents to the auctioneer whenever they want to submit a bid, which is forbidden in the proposed scheme.

This being said, the agents must affect the global signal in some way, as the signal is used to satisfy the equality constraint, which depends on the agents behaviors. Indeed, the information loop is closed by the results of the agents actions on the physics underling the problem. The collective effect can be observed (either directly by each agent or indirectly through the announcer) and this provides sufficient information to satisfy the constrained parallel production problem. The added benefit of this physical communication is that cheating or deceitful behaviors are ruled out by design. An agent can not communicate the intent of an action and then act in a different way; the action *is* the message.

Finally, we highlight some less technical benefits related to system and component design. Similarly to the non-cooperative approach, we can strongly decouple system and component behavior, but the control schemes are much simpler.

After a more formal problem formulation, we will investigate more closely the decentralized constrained optimization to show how dual ascent can satisfy the requirements presented so far. The method of multiplier and the alternating method of multipliers on the other hand, require some exchange of information between the agents and thus violate the no-communication constraint.

2.3 Formal model definition

2.3.1 Individual agent

A simple model is used to represent the agent contribution to production; each agent is viewed as single input-single output production plant, as depicted in figure 2.2. Since the agent only task is the production of one good, it is convenient to identify its production level x as the only state variable for the agent[8] neglecting any internal complex dynamics. For agent j then, $x_j(t)$ identifies both its state at time t and the amount of good produced at t. Without loss of generality, we normalize the production of each agent and assume that $x \in [0, 1]$. The function that maps good production to the amount of resource consumed is $g(x)$. Hence, the efficiency of the individual agent is defined as $\varepsilon_j(x) = \frac{x}{g_j(x)}$. An example of $\varepsilon_j(\cdot)$ and $g_j(\cdot)$ is shown in figure 2.3. Note that while the efficiency curve might be quite general, consumption can be always though as monotonically increasing with production.

The objective is to provide an algorithm that enables each agent to independently chose the most appropriate production level in order to meet the request and maximize cluster efficiency.

[8] This is just an expedient; one could use x as the internal state of the agent and then have another function $f(\cdot)$ to map the internal state to the output. As long as the function f admits an inverse, the two methods are equivalent.

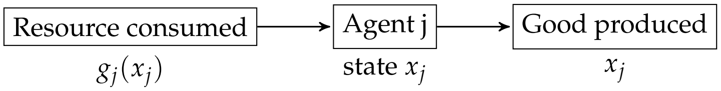


Figure 2.2: Single input-Single output model for a generic agent

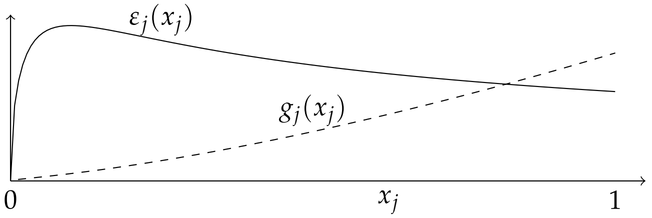


Figure 2.3: Example of efficency curve (smooth line) and cost curve/resource consumption curve (dashed)

2.3.2 Cluster

Consider the situation shown in Fig. 2.4, where a cluster of n single input-single output agents partakes in the same production problem. Due to the parallel architecture, the cluster output will be the sum of the output of the agents, respectively $x_1, ..., x_n$. The cluster input, the amount of resources that the cluster consumes to produce the output, will be the sum of the consumption of each agent, respectively $g_1(x_1), ..., g_n(x_n)$. Formally, for the cluster we can define

$$\text{Cumulative Production} \doteq \sum_{i=1}^{n} x_i \qquad \text{Cumulative Consumption} \doteq \sum_{i=1}^{n} g_i(x_i) \qquad \mathcal{E}_c \doteq \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} g_i(x_i)} \tag{2.3}$$

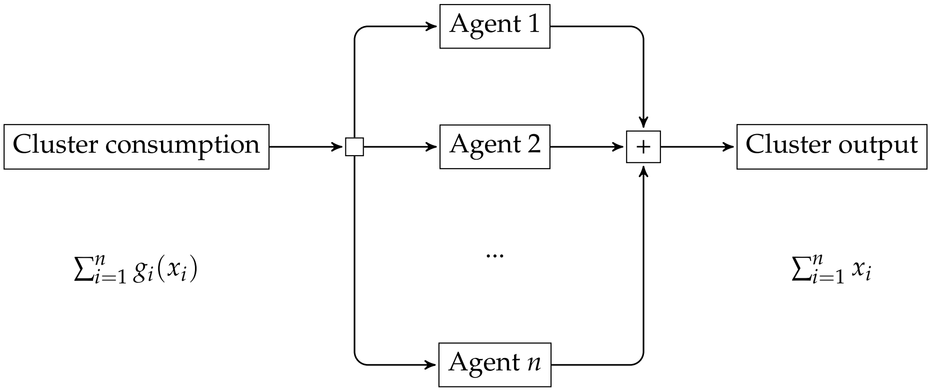


Figure 2.4: A cluster of n *agents* with parallel consumption and production.

To satisfy constrained optimization in a decentralized environment we propose that each agent in the cluster follows the law:

$$\dot{x}_i = k_0 \left(R - \sum_{j=1}^{n} x_j \right) - \frac{\partial g_i}{\partial x_i} \qquad i = 1, 2, \ldots, n \tag{2.4}$$

Where R is the amount of good requested, k_0 some positive constant. In words, Eq. 2.4 states that the adjustment that each agent makes to its current output is largely proportional to the total production error ($R - \sum_{j=1}^{n} x_j$), but also accounts for the effect that changing its output will have on its efficiency.

Note that the only external information needed by the ith-agent is $R - \sum_{j=1}^{n} x_j$, which has the same value for every agent and thus can be viewed as a global variable. We will show that the control law 2.4 guarantees that

1. The cluster *quickly* reaches the target request R
2. At equilibrium, cluster consumption is minimized

2.3.3 Dual ascent

Thanks to the nature of the problem, Dual Ascent can be implemented under the same constraint as the proposed method; namely without direct communication between the agents. We therefore review the method more in detail as it will be used as a benchmark in the next chapter, where we numerically characterize the performances of the proposed algorithm.

It is important to remark that proof of convergence for this method are known only for convex cost functions. The lagrangian of problem 2.1 can be written with the nomenclature introduced so far as

$$\mathcal{L}(x, y) = \sum_{j=1}^{n} g_j(x_j) + y(R - \sum_{j=1}^{n} x_j) \tag{2.5}$$

Where y is the Lagrange multiplier, or dual variable. The algorithm at the $k + 1$th step can be written as

$$\begin{cases} x^{k+1} & = & \underset{x}{\operatorname{argmin}}\, \mathcal{L}(x, y^k) \\ y^{k+1} & = & y^k + \alpha^k \cdot \nabla_y \mathcal{L} \end{cases} \tag{2.6}$$

Where α is a suitably small constant that regulates the gradient ascent step of the y variable. Since the objective function $\sum_{i=1}^{n} g_i(x_i)$ is naturally decoupled in x_i, the update step does not need communication. Each agent i can update its output for the next step according to

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}} \left(\sum_{j=1}^{n} g_j(x_j) + y^k(R - \sum_{j=1}^{n} x_j) \right) \tag{2.7}$$

Assuming that the $g_i(x_i)$ functions are convex, we can solve *argmin* analytically by setting the derivative to zero

$$\frac{\partial \mathcal{L}}{\partial x_i} = 0 \quad \Leftrightarrow \quad \frac{\partial g_i(x_i)}{\partial x_i} = y^k \tag{2.8}$$

and therefore, only local information are needed to update x_i.

2.3.4 Method of multipliers

In the case in which the g_i are not strictly convex, we can augment the Lagrangian by adding a quadratic term which does not alter the constrained optima

$$\mathcal{L}^+(x,y) = \sum_{i=1}^{n} g_i(x_i) + \frac{\rho}{2} \cdot \left(R - \sum_{i=1}^{n} x_i\right)^2 + y \cdot \left(R - \sum_{i=1}^{n} x_i\right) \tag{2.9}$$

where $\rho \in \mathbb{R}^+$. For a suitably large value of ρ, $\mathcal{L}^+$ can be made convex, but its gradient is no longer separable. In order to perform the *argmin* step, each agent needs access to the full state of the system. The method of multipliers is not distributed.

$$\frac{\partial \mathcal{L}^+}{\partial x_i} = \frac{\partial g_i(x_i)}{\partial x_i} - y - \rho\Big(R - \sum_{j=1}^{n} x_i\Big) \tag{2.10}$$

The method of Alternating Direction Method of Multipliers solves this problem by using the states from the previous step; however some communication scheme is required to redistribute individual states, which is against problem statement. Therefore, only Dual Ascent is capable of achieving decentralization without communication.

Toy problem with n=2

To introduce the method, we consider the case where there are only 2 agents. All functions describing the cluster allocation will therefore be defined within a subset of $\mathbb{R}^2$. To further embrace this intuitive representation, we will call x the output produced by the first agent/component and y the output of the second one. We will use this toy model to provide visual geometric intuition about the statements that might otherwise seem unnecessarily complex.

The control law implemented by each agent is:

$$\begin{cases} \dot{x} &= k_0 \cdot (R - x - y) &- &\frac{\partial g_1}{\partial x}(x) \\ \dot{y} &= k_0 \cdot (R - x - y) &- &\frac{\partial g_2}{\partial y}(y) \end{cases} \tag{2.11}$$

and we will prove that it satisfies all requirements, namely it converges to target request, it minimizes consumption and it does not require communication between the two agents.

General idea for convergence to target request

To intuitively understand why the system converges to $x_{t_\infty} + y_{t_\infty} \approx R$, notice that consumption functions g_1, g_2 of **real components** are:

- Defined over a finite and closed domain
- Bounded; any output that we can produce will require only finite resources
- Smooth; $g_x, g_y \in \mathcal{C}^0$

Therefore, we can say that

$$G_x \doteq \max_{x\in[0,1]} g_x(x) < +\infty \qquad G_y \doteq \max_{y\in[0,1]} g_y(y) < +\infty$$

Since the gain k_0 in Eq. 2.11 is a degree of freedom for the control design, if we choose $k_0 >> \max(G_x, G_y)$, we can approximate the dynamic of the cluster (2.11) with a simple linear model

$$\begin{cases} \dot{x} & \approx & k_0 \cdot (R - x - y) \\ \dot{y} & \approx & k_0 \cdot (R - x - y) \end{cases} \tag{2.12}$$

Then, if we define $\Delta R \doteq R - x - y$, $\dot{\Delta R} = \dot{R} - \dot{x} - \dot{y}$. Assuming $\dot{R} \approx 0$ (request is constant, or at least has a dynamic which is much slower than those of $\dot{x}, \dot{y}$), we can re-write system (2.12) as

$$-\dot{x} - \dot{y} = \dot{\Delta R} = -k_0 \cdot \Delta R - k_0 \cdot \Delta R \quad \Rightarrow \quad \Delta R(t) = \Delta R_0 \cdot e^{-2k_0 t}$$

Then, as long as k_0 is positive, we have a very *sturdy* dynamics, which quickly reaches $\Delta R = 0 \Rightarrow R = x + y$, regardless of initial conditions (x_0, y_0).

The above argument should convince us that it is possible to achieve the target R with some accuracy and that we can do so in a distributed way. We now focus on efficiency maximization.

General idea for consumption minimization without communication

By choosing a suitably large gain k_0, we expect linear dynamics to force the system into an arbitrarily small region *around* the set of points where $\Delta R = 0$. Here, the gradient dynamic will become dominant.

To simplify things, assume $\Delta R \approx 0 \Rightarrow x + y \approx R$. Then, efficiency can be re written as

$$\varepsilon_c = \frac{x+y}{g_1(x) + g_2(y)} \approx \frac{R}{g_1(x) + g_2(y)}$$

This leads to a separable but equivalent version of the second objective

$$\max_{\Delta R \approx 0} \frac{R}{g_1(x) + g_2(y)} \quad \Leftrightarrow \quad \min_{\Delta R \approx 0} g_1(x) + g_2(y)$$

Therefore, if the linear dynamic guarantees that $\Delta R \approx 0$, the maximization problem becomes separable. There is no need for communication. Cluster maximization of efficiency will be achieved by the agent minimization of individual consumption. Under the assumption that $\Delta R \approx 0$, the dynamic becomes

$$\begin{cases} \dot{x} & = & - & g_x(x) \\ \dot{y} & = & - & g_y(y) \end{cases} \tag{2.13}$$

Clearly, this is a gradient system, for which all local minima of $g_{tot}(x, y) = g_1(x) + g_2(y)$ are stable.

2.4 Formal results statement

Theorem 1 (Continuous Time)**.** Consider a cluster of n agents, each with its own consumption function $g_i : [0, 1] \to \mathbb{R}^+$ and $g_i \in \mathcal{C}^2([0, 1])$. Then, there exists $\bar{k} > 0$ such that $\forall\, k_0 > \bar{k}$, given the control law

$$\dot{x}_i = k_0 \left(R - \sum_{j=1}^{n} x_j \right) - \frac{\partial g_i}{\partial x_i} \qquad i = 1, 2, \dots, n$$

the cluster dynamic will exhibit the following properties

$$\lim_{t \to +\infty} \left\| R - \sum_{i}^{n} x_i(t) \right\| < \frac{1}{k_0} \cdot \max_{X \in \mathcal{X}} \left\| \frac{\partial G}{\partial x_i} \right\| \tag{2.14}$$

$$\lim_{t \to +\infty} G(X(t)) \leq G(X) \quad \forall\, X \in \left\{ X \text{ such that } \sum_{i=1}^{n} x_i = R \right\} \cap \mathcal{B}_\delta(X(t_\infty)) \tag{2.15}$$

Where $X = (x_1, \dots, x_n)^\intercal$ and $\mathcal{B}_\delta(X)$ is a ball of radius δ centered in X.

Intuitively, the idea is for k_0 to be a sufficiently large positive constant. From Eq 2.14, the larger it is, the smaller the error in total production will be. The subscript $_0$ is used to set it apart from the usual iterator index k, but can also suggest an iterative process to further optimize the cluster behavior. We can provide a bound for the value of $\bar{k}$

Theorem 2. We can provide an estimate of $\bar{k}$ as

$$\bar{k} \leq \frac{\| \min\limits_{i=1,..,n} (g_{xx_i}) \|}{n} \tag{2.16}$$

Although we do not know, in general, the exact value $\bar{k}$ (which acts as a lower limit) if you choose a k_0 bigger than the upper estimate provided by theorem 2 (above), theorem 1 holds.

Since the system is dynamic, the solution (equilibrium) is reached after a certain time. In practical circumstances it might be very important to characterize the speed at which target production is reached. Fortunately, convergence is quite fast, as stated by the following result.

Theorem 3 (Speed)**.** The cluster approaches the requested production level as

$$\Delta R(t) \approx \Delta R_0 e^{-nk_0 t} \qquad \Delta R \gg 1 \tag{2.17}$$

More refined characterization might be obtained, as the Lyapunov functions we will use in the proofs are quadratic; however the author was not able to obtain stronger results.

2.5 Proofs

The proofs are organized according to the following road map

- Nomenclature and basic observations
- Proof that all equilibrium points are stationary points for the consumption function using invariance principle/LaSalle.

- Proof that minima consumption points are stable using Lyapunov.
- Proof that maxima and saddle points are unstable using Chetaev
- Speed Characterization

To study the overall cluster behavior, it is convenient to use the state space approach . The set of differential equations that govern the system is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dots \\ \dot{x}_n \end{pmatrix} = k_0(R - \sum_{i=1}^{n} x_i) \cdot \begin{pmatrix} 1 \\ 1 \\ \dots \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{\partial g_1}{\partial x_1} \\ \frac{\partial g_2}{\partial x_2} \\ \dots \\ \frac{\partial g_n}{\partial x_n} \end{pmatrix} \tag{2.18}$$

We can re write equation 2.18 introducing the vectorial equivalent. We call $X \in [0,1]^n \subset \mathbb{R}^n$ the vector of individual output $X(t) \doteq (x_1(t), x_2(t), \dots, x_n(t))^\intercal$, and $G(X) : \mathbb{R}^n \to \mathbb{R}$ the cumulative consumption function $G(X) \doteq \sum_{i=1}^{n} g_i(x_i)$

$$\frac{\partial}{\partial t} X(t) = f(X(t)) \qquad f(X(t)) = k_0(R - \vec{1}^\intercal \cdot X(t)) \cdot \vec{1} - \nabla G(X) \tag{2.19}$$

Where $\vec{1} = (1, 1, \dots, 1)^\intercal$.

2.5.1 Proof of convergence to stationary points along the constraints

We want to show that every equilibrium point (whether stable or not) is arbitrarily close to the constraint and represents a stationary point for the efficiency (derivative is zero in the direction tangent to the constraint).

To use the invariance principle (Th. 5), we must guarantee that cluster dynamics does not exit a compact set, which we choose as the whole domain $\chi = [0,1]^n$. Therefore for any $t > 0$, $X(t) \in \chi$. This has a immediate practical meaning. Each component of the X vector represents the amount of output for each agent. Agents can be either completely turned off (output is $0 = x_i$), completely turned on (maximum output $1 = x_i$), or anything in between. It would not make sense to allow a component to output more than its maximum capacity; the mathematical model would diverge from the engineering one. This feature can either be a natural propriety of the cluster [9] or can be artificially implemented. To artificially set boundaries, we can limit the derivatives components of X as[10]

$$\dot{x}_i|_{\text{corrected}} = \begin{cases} \text{if} \quad \dot{x}_i > 0 \quad \text{and} \quad x_i = 1 \quad \Rightarrow \quad 0 \\ \text{if} \quad \dot{x}_i < 0 \quad \text{and} \quad x_i = 0 \quad \Rightarrow \quad 0 \\ \qquad\qquad\quad \text{else} \qquad\qquad\qquad\quad \dot{x}_i \end{cases} \tag{2.20}$$

Smoother switches can be devised in order to guarantee continuity of the time derivatives on the boundary of the domain, however these are enough for this proof.

[9] By not requesting $R > n$, and in the case that both components become more inefficient as more output is produced

[10] This works only in continuous system, for a discrete time implementation, we need a more careful choice of parameters

As a Lyapunov function, we choose $V(X)$ such that its derivative with respect to time is a simple quadratic function

$$\frac{\partial V}{\partial t} \doteq - \dot{X}^{\intercal} \cdot \dot{X} = - \sum_{i=1}^{n} \dot{x}_i^2 \tag{2.21}$$

Which guarantees that $\dot{V} \leq 0$ for all $X \in \chi$ and that, for every $X \in \chi$ such that $\dot{V} = 0$, all components of $\dot{X}$ must be zero.

$$\frac{\partial V}{\partial t} = 0 \quad \Leftrightarrow \quad \dot{x}_i = 0 \quad \forall i = 1, 2, ..., n \tag{2.22}$$

With some manipulation we can write a $V()$ that satisfies the differential Eq. 2.21 . Since

$$\frac{\partial V}{\partial t} = \nabla V \cdot \dot{X} = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \cdot \dot{x}_i \tag{2.23}$$

We can satisfy Eq. 2.21 by imposing that $\frac{\partial V}{\partial x_i} = -\dot{x}_i$. It is easy to check that Eq. 2.24 satisfies all the requirements

$$V(X) \doteq - k_0 R \cdot \vec{1}^{\intercal} X + G(X) + \frac{k_0}{2} \cdot X^{\intercal} X \tag{2.24}$$

Which is $C^1(\chi, \mathbb{R})$ since $G \in C^2(\chi, \mathbb{R})$. Then we have satisfied all the hypothesis of the invariance principle, and therefore $X(t)$ will converge to the largest positively invariant set $\mathcal{M}$ contained in $\mathcal{Z} = \{X \in \chi \text{ such that } \dot{V}(X) = 0\}$.

A proper characterization of $\mathcal{M}$ is not necessary for this weak proof. It will be sufficient to notice that, for any point of $\mathcal{M} \subset \mathcal{Z}$,

$$\dot{V}(X) = 0 \Leftrightarrow \dot{X} = \vec{0} \quad \Rightarrow \quad k_0 \cdot (R - \vec{1}^{\intercal} X) = \frac{\partial G}{\partial x_i} \quad \Rightarrow \quad R - \vec{\mathbb{1}}^{\intercal} X = \frac{1}{k_0} \cdot \frac{\partial G}{\partial x_i} \tag{2.25}$$

Noticing that $R - \vec{1}^{\intercal} X$ is the error in achieving the production R, we understand that by choosing a suitably large k_0, all the points in the $\mathcal{Z}$ set can be pushed arbitrarily close to $\mathrm{E}rr(R) = 0$. This allows an estimation of the maximum error we can expect on the first objective

$$|\mathrm{Err}| \leq \frac{\max\limits_{X \in [0,1]^n} \left\{ \| \frac{\partial G}{\partial x_i} \| \quad i = 1, 2, .., n \right\}}{k_0} \tag{2.26}$$

Furthermore, only stationary points along the constraint can be part of $\mathcal{Z}$. We can show that, on any point $\Omega \in \mathcal{Z}$, the derivative of the consumption function $G(X)$ in any direction $\vec{s}$ that is tangent to the constraints, must be zero. The directional derivative can be written as

$$D_{\vec{s}} G(\Omega) = \nabla G(\Omega) \cdot \vec{s} = \sum_{i=1}^{n} \left. \frac{\partial G}{\partial x_i} \right|_{\Omega} \cdot s_i = \left. \frac{\partial G}{\partial x} \right|_{\Omega} \cdot \left(\sum_{i=1}^{n} s_i \right) = 0 \tag{2.27}$$

First we have used the fact that all derivatives $\frac{\partial G}{\partial x_i}$ evaluated in Ω assume the same value (as shown in the middle part of Eq. 2.25). Finally, since moving in the direction $\vec{s}$ does not change total production[11], it must mean that $\sum_{i=1}^{n} s_i = 0$.

[11] By definition $\vec{s}$ is tangent to the constraint that keeps total production constant.

Hence any Ω in $\mathcal{Z}$ is a constrained stationary point. This result is weak because it does not discern between maxima, minima or saddle points. However, it provides interesting global information, proving that $X(t)$ converges to stationary points, regardless of $X(t=0)$.
In the next sections, we will first show that minima are locally stable points and finally that maxima and saddle are unstable.

2.5.2 Proof of local stability of constrained minima

For this proof, we will use the classical Lyapunov stability theorem (Th. 4). We choose a different Lyapunov function, quadratic with the derivative.

$$V(X) = \left(\frac{\partial}{\partial t}X\right)^2 = f(X)^2 \tag{2.28}$$

Its derivative is then

$$\frac{\partial}{\partial t}V(X) = (2 \cdot f(X) \cdot \nabla f(X)) \cdot \frac{\partial}{\partial t}X = 2f(X) \cdot \nabla f(X) \cdot f(X)$$

Where, since $\nabla f(X)_{i,j} = \frac{\partial}{\partial x_j} f(X)_i$ we have

$$\nabla f(X) = -\begin{bmatrix} k_0 + \frac{\partial^2 G}{\partial x_1^2} & k_0 & \dots & k_0 \\ k_0 & k_0 + \frac{\partial^2 G}{\partial x_2^2} & \dots & k_0 \\ k_0 & k_0 & \dots & k_0 + \frac{\partial^2 G}{\partial x_n^2} \end{bmatrix} \tag{2.29}$$

Our goal is to prove that an equilibrium point Ω is stable if it is a constrained minima for G. To prove the stability, we need to show that $\dot{V}$ is negative in a closed set around Ω, which means proving that the matrix $-[\nabla f(X)]$, which we will rename $\mathbf{A}$ for convenience, is definite positive around Ω. By proving that $\mathbf{A}$ is definite positive in Ω, we can find, by continuity a small ball in which it is still definite positive.

Assuming Ω is a minimum for G along the constraint, its value must increase along any path admissible by the constraint, meaning any path that preserves the total output. More formally, the second derivative evaluated in any direction $\vec{s}$ on the hyperplane of equi-production must be positive. This means that

$$\forall \vec{s} \text{ such that } \vec{s} \cdot \begin{pmatrix} 1 \\ 1 \\ \dots \\ 1 \end{pmatrix} = 0, \quad \nabla^2 G(\Omega)|_{\vec{s}} = \vec{s}^{\mathsf{T}} \begin{bmatrix} g_{xx_1} & 0 & \dots & 0 \\ 0 & g_{xx_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & g_{xx_n} \end{bmatrix} \vec{s} > 0$$

Which can be written component by component as

$$\forall s_1, s_2, \dots, s_n \in \mathbb{R} \text{ such that } \sum_{i=1}^{n} s_i = 0, \quad \sum_{i=1}^{n} s_i^2 \cdot g_{xx_i} > 0 \tag{2.30}$$

Therefore we can say that at most one g_{xx_i} can be negative. To prove it, we rearrange the terms of g_{xx_i} in increasing order as $g_{xx\ \mathrm{min}} \doteq g_{xx_p} < g_{xx_j} < g_{xx_k} < \cdots < g_{xx_t} \doteq g_{xx\ \mathrm{max}}$ and choose the vector $\vec{s}$ as $s_p = -s_j,\ s_i = 0\,\forall i \neq p, j$. Then we have

$$s_p^2 \cdot (g_{xx_p} + g_{xx_j}) > 0 \quad \Rightarrow \quad g_{xx_j} > 0$$

So there can not be more than one non positive g_{xx_i}, and if there is a negative g_{xx_j}, its absolute value is smaller than the minimum non negative g_{xx}. We have proven that, in Ω, it holds that

$$g_{xx_p} < g_{xx_j} \leq g_{xx_m} \leq \cdots \leq g_{xx_l} \qquad g_{xx_p} + g_{xx_j} > 0 \tag{2.31}$$

We claim that, as long as condition 2.31 applies, the matrix $\mathbf{A}$ can be made definite positive by a suitably large k_0. To use the definition of definite positive, we need to prove that

$$\forall v \in \mathbb{R}^n \setminus \vec{0} \qquad v^\intercal \mathbf{A} v = v^\intercal (k_0 \mathbb{1} + \nabla G) v = k_0 \cdot v^\intercal \mathbb{1} v + v^\intercal \nabla G v > 0 \tag{2.32}$$

Where $\mathbb{1}$ is the matrix with 1 on every entry. Without loss of generality, we can consider v such that $\|v\| = 1$. For for all v on the constraint, $v^\intercal \nabla G v > 0$, while $\vec{1}^\intercal v = \vec{0}$. Therefore any value of k_0 will work.

Hence, we only need to consider the direction normal to the constraint. In this direction $v^\intercal \mathbb{1} v = n\|v\|$, and the second derivative could be negative.

$$v^\intercal \mathbf{A} v \geq n k_0 \|v\| + \min_{v \in \mathbb{R}^n} (v^\intercal \nabla G v) = n k_0 + \min_{i=1,..,n} (g_{xx_i}) > 0 \quad \Leftarrow \quad k_0 > \frac{\|\min_{i=1,..,n} (g_{xx_i})\|}{n} \tag{2.33}$$

This is a sufficient condition. Interestingly, the requirements on k_0 get less stringent with n, as more agents join the problem.

Then, we can expand the positive definite quality of the matrix $\mathbf{A}$ to a neighborhood $\mathcal{N}_\varepsilon(\Omega)$ by expanding the validity of condition 2.31 by continuity. Since we have assumed $G \in \mathcal{C}^2(\chi)$, all the second derivatives continuous, $g_{xx_i} \in \mathcal{C}^0(\chi)$ and therefore there will exist a neighborhood $\mathcal{N}_\varepsilon(\Omega)$ in which condition 2.31 is still true.
This proves that, if Ω is a constrained minimum, there exists δ such that $\dot{V} < 0 \in \mathcal{N}_\delta(\Omega)$. By Lyapunov then, Ω can be stabilized, regardless of the convexity of G and regardless of the number of agents in the cluster, by an appropriate choice of k_0.

2.5.3 Instability of maxima and saddle points

If we prove that the maxima (and saddle) points are unstable equilibria, we have that the cluster dynamic rejects these points, thus converging to the minima. We will use Chetaev theorem (in appendix Th. 6). We need to show that, for some appropriate $V \in \mathcal{C}^1(\chi)$, there is (at least) a continuous path characterized by $V > 0$ and $\dot{V} > 0$ that starts in the stationary point Ω and exits the neighborhood $\mathcal{N}_r(\Omega)$. Our V function is a distance from the equilibrium point X_Ω, which is always non negative except in X_Ω, where it is 0

$$V(X) = (X - X_\Omega)^2 \qquad \frac{\partial V}{\partial t} = 2(X - X_\Omega)(f(X) - f(X_\Omega)) \tag{2.34}$$

Since X_Ω is an equilibrium point and considering a small movement $\vec{s} = X - X_\Omega$ we can write (using Taylor approximation)

$$\frac{\partial V}{\partial t} = 2\vec{s} \cdot (\cancel{f(X_\Omega)} + \nabla f(X_\Omega) \cdot \vec{s} - \cancel{f(X_\Omega)}) = -2\vec{s}(\mathbb{1}k_0 + \nabla^2 G(X_\Omega))\vec{s} \tag{2.35}$$

Considering $\vec{s}$ in the constraint direction $\sum_{i=1}^{n} s_i = 0$, we have that

$$\frac{\partial V}{\partial t} = -2\vec{s}\,\cancel{\mathbb{1}\vec{s}}k_0 - 2\vec{s}\,\nabla^2 G(X_\Omega)\vec{s} = -2\vec{s}\,\nabla^2 G(X_\Omega)\vec{s} \tag{2.36}$$

If X_Ω is a maximum along the constrained direction, $\vec{s}^{\intercal}\,\nabla^2 G\vec{s} < 0 \Rightarrow \dot{V} > 0$. Hence, in the neighborhood of a constrained maxima there is a clear escape route along the constraint. Moving in any direction $\vec{s}$ on the constraint will increase the value of V, and retain a positive $\dot{V}$.

If X_Ω is a saddle point, there must be at least one direction in which the product $\vec{s}\,\nabla^2 G\vec{s}$ is negative, otherwise it would be constrained minima. Therefore the same reasoning applies, but not in all directions on the constraint.

2.5.4 Speed characterization

We want to obtain some sort of weak characterization for the speed at which the algorithm converges toward the constraint. First, notice that the system has two, well separated dynamics and objectives; to reach the target output and to minimize consumption. To characterize the speed to which the first is accomplished is very easy, and it is perhaps more interesting. In fact

$$\Delta R = R - \vec{1}^{\intercal} X \quad \Delta\dot{R} = -\vec{1}^{\intercal}\dot{X} = -nk_0 \cdot \Delta R + \sum_{i=1}^{n} \frac{\partial G}{\partial x_i} \tag{2.37}$$

Since we are choosing k_0 to be big (at least $>$ max $\|\frac{\partial G}{\partial x_i}\|$ for stability, bigger to reduce the constraint error), we can approximate the derivative for large values of ΔR, hence

$$\Delta\dot{R} \approx nk_0 \cdot \Delta R \quad \rightarrow \quad \Delta R(t) = \Delta R_0 e^{-nk_0 t} \quad \text{for } \Delta R \gg 1 \tag{2.38}$$

Therefore, the dynamic towards the constraint is, to a good approximation a proportional one until an acceptable approximation is reached. At that point, the optimization dynamics becomes dominant.

Appendix; Fundamental Theorems in Nonlinear Control

To support the reader, we recap the fundamental results of non linear control theory used in this chapter. For the notation, we will refer to the general time invariant system

$$\dot{x} = f(x), \quad x(0) = x_0 \tag{2.39}$$

with f such that a solution $x(t)$ exists unique in $\chi \subset \mathbb{R}^n$ for any $x_0 \in \chi$.

Theorem 4. (Lyapunov stability) Let $\bar{x} = 0 \in \mathcal{D}$ be an equilibrium, and $\mathcal{D}$ an open connected set. Assume $V \in \mathcal{C}^1(\mathcal{D}, \mathbb{R})$ is such that

1. $V(0) = 0$ and $V(x) > 0$ for $x \in \mathcal{D} \setminus 0$
2. $\dot{V} \leq 0$ for $x \in \mathcal{D} \setminus 0$

Then $\bar{x} = 0$ is stable for the system 2.39. If additionally

$$\dot{V}(x) < 0 \qquad \text{on } \mathcal{D} \setminus 0$$

then $\bar{x}$ is asymptotically stable.

Definition 2.5.1. (Invariant Set) A subset $\mathcal{S}$ of χ is said to be (positively) invariant if

$$x(0) \in \mathcal{S} \Rightarrow x(t) \in \mathcal{S}, \forall t > 0$$

Note that an invariant set can have non trivial internal dynamics, while a set of invariant states does not.

Theorem 5. (Invariance Principle- Barbashin-Krasowskii/LaSalle) Assume that there exist $\Omega_0 \subset \chi$ such that, for all $x_0 \in \Omega_0$, the solution does not exit a compact $\Omega \subset \chi$. Let $V \in \mathcal{C}^1(\chi, \mathbb{R})$, and assume $\dot{V}(x) \leq 0$ for $x \in \Omega$. Call $\mathcal{Z} = \{x \in \Omega \text{ such that } \dot{V}(x) = 0\}$, and $\mathcal{M}$ the largest positively invariant set in $\mathcal{Z}$. Then $\Phi(x_0)$ converges to $\mathcal{M}$ for all $x_0 \in \Omega_0$.

Theorem 6. (Chetaev) Let $(x, y) = \Omega$ be an equilibrium point for the system and assume $V : \mathcal{C}^1(\chi, \mathbb{R})$ to be such that for any $\varepsilon > 0$ there exists $(x, y) \in \mathcal{N}_\varepsilon(\Omega)$ that satisfies $V(x, y) > 0$. Let $r > 0$ be such that $\mathcal{N}_r(\Omega) \subset \chi$ and define

$$U_r = \{(x, y) \in \mathcal{N}_r(\Omega) \text{ such that } V(x, y) > 0\}$$

If $\dot{V}(x, y) > 0$ for all $(x, y) \in U_r$, then Ω is unstable.

Theorem 7. Let $f(x)$ be a locally Lipschitz function defined ovar a domain $D \subset \mathbb{R}^n$, which constains the origin, and $f(0) = 0$. Let $V(x)$ be a continuously differentiable function defined over D such that

$$k_1 \|x\|^\alpha \leq V(x) \leq k_2 \|x\|^\alpha \tag{2.40}$$

$$\dot{V}(x) \leq -k_3 \|x\|^\alpha \tag{2.41}$$

for all $x \in D$, where k_1, k_2, k_3, α are all positive constants. Then, the origin is an exponentially stable equilibrium point of $\dot{x} = f(x)$. If the assumptions hold globally, the origin will be globally exponentially stable.

Bibliography

[1] Yoav Shoham, Kevin Leyton-Brown, *Multiagent Systems:Algorithmic, Game-Theoretic, and Logical Foundations* http://www.masfoundations.org

[2] R. G. Smith, *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver*, *IEEE*, 1980.

[3] Tuomas Sandholm, *An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations, AAAI 1993*

[4] J. Wu, *Contract Net Protocol for Coordination in Multi-agent System*, *IEEE, 2008*

[5] Vincent D.Blondel, Julien M.Hendrickx, Alex Olshevsky, Jhon N. Tsitsiklis, *Convergence in Multiagent Coordination, Consensus, and Flocking*, *IEEE*, 2005.

[6] R.Carli, F. Fagnani, A. Speranzon, S.Zampieri, *Communication constraints in coordinated consensus problems*, *IEEE*, 2006

[7] R. Olfati,R. Saber, M. Murray *Consensus Problems in Networks of Agents with Switching Topology and Time- Delays IEEE Trans on Automatic Control, Vol: 49 , Sept. 2004*

[8] W. Ren, R.W. Beard, *Consensus seeking in multiagent systems under dynamically changing interaction topologies*, *IEEE Trans. on Automatic Control, Vol: 50 , May 2005*

[9] Duŝan Jakovetić, Joá o X. Moura J.M.F, *Fast Distributed Gradient Mehods*, XXXX 2014

[10] J.N. Tsitsiklis, S. Bertsekas,M. Athans, *Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms*, *IEEE*, 1986

[11] K. Yuan, Q. Ling, W. Yin, *On the convergence of Decentralized Gradient Descent*, X?XX 2015

[12] Angelia Nedić, Asuman Ozdglar, *Distributed Subgradient methods for Multi-Agent Optimization*, *IEEE*, 2009

[13] Angelia Nedić, Asuman Ozdaglar, and Pablo A. Parrilo, *Constrained Consensus and Optimization in Multi-Agent Networks*, *IEEE trans. Autom. Control.*,Vol. 55 no. 4, 2010

[14] Minghui Zhu, Sonia Martínez *On Distributed Convex Optimization Under Inequality and Equality Constraints IEEE trans. Autom. Control.*,Vol. 57 no. 1, 2012

[15] Y.Cao, W. Yu, W. Ren, and G. Chen, *An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination IEEE Transactions on Industrial Informatics, 2012*

[16] William Vickrey, *Counterspeculation, Auctions, and Competitive Sealed Tenders*, The Journal of Finance, Vol. 16, No. 1 (Mar., 1961)

[17] J. Marden, G. Arslan, and J. Shamma, *Cooperative Control and Potential Games*, *Trans. on System, Man, and Cybernetics VOL. 39, NO. 6, Dec 2009*

[18] N. Li, J. Marden, *Designing Games for Distributed Optimization IEEE, J. of Selected Topics in Signal Processing, Vol. 7, No. 2, April 2013*

[19] J. Zhang, D. Qi, M. Yu, *A Game Theoretic Approach for the Distributed Control of Multi-Agent Systems under Directed and Time-Varying Topology International Journal of Control, Automation, and Systems (2014)*

Chapter 3

Application to small sat ADCS

We now turn our attention to a more concrete setting. From the generic problem, with generic actuators and abstract cost functions, we focus on a common actuator for ADCS in small satellites, reaction wheels (RW). We chose RWs because they are the dominant mean of attitude control,they are quite simple to model analytically and to reproduce with minimal hardware set up. In this chapter we will:

1. Review the analytical model for a DC engine.
2. Provide an empirical model derived from HW data
3. Discuss the role of the cost function
4. Compare various optimization strategies and their effectiveness

3.1 Analytical model of a RW

We model the behavior of a RW as a standard DC motor. From the coupled electrical and mechanical model (as shown in Fig. 3.1) we obtain the following equations

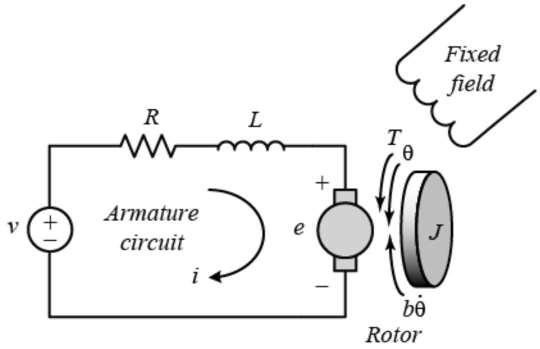


Figure 3.1: The standard electro-mechanical model for a DC engine

$$V(t) = R \cdot i(t) + L\frac{\mathrm{d}\,i(t)}{\mathrm{d}\,t} + E(t) \tag{3.1}$$

$$E(t) = k_v \cdot \omega(t) \tag{3.2}$$

$$T_m(t) = k_t \cdot i(t) \tag{3.3}$$

$$T_m(t) = J\frac{\mathrm{d}\omega}{\mathrm{d}t} + B\omega(t) + T_f \tag{3.4}$$

Where $V(t)$ is the applied potential, $E(t)$ is the electromotive force, J is the inertia of the rotor, B is the coefficient for linear friction and T_f is the non linear friction function. The parameter R, L, k_v, k_t and B are characteristics of the motor.

To help the intuition, when plotting relationships for the analytical model, we will use the motor parameter in table 3.1, which refers to a somewhat large RW for CubeSat.

Parameter	Value	Unit
R	2	Ω
L	$2{\cdot}10^{-4}$	H
k_v	0.01	V/(rad/s)
k_t	0.01	Nm/A
B	$3 \cdot 10^{-8}$	Nm/(rad/s)
J	$1 \cdot 10^{-4}$	Nm/(rad/s^2)
ω_{max}	840	rad/s

Table 3.1: Reference values for a RW

Note that, typically, $\frac{L}{R}$ is small ($\approx 10^{-4}$), therefore Eq. 3.1 can be reduced to

$$V(t) = R \cdot i(t) + E(t)$$

3.1.1 Power consumption model

Reaction wheels are used to apply a torque to the satellite. Therefore an appropriate model within the single input - single output framework would be an actuator that consumes electrical power and produces a torque. We wish to derive a function like

$$P(T_{\text{out}}, \omega) = V(T_{\text{out}}, \omega) \cdot i(T_{\text{out}}, \omega) \tag{3.5}$$

Using Eq 3.1-3.4, we can write

$$\begin{aligned} V(T_{\text{out}}, \omega) &= R \cdot i(T_{\text{out}}, \omega) + k_v \cdot \omega \\ i(T_{\text{out}}, \omega) &= \frac{1}{k_t} \cdot (T_{\text{out}} + B\omega) \end{aligned}$$

where we have defined $T_{\text{out}} \doteq J\frac{\mathrm{d}\omega}{\mathrm{d}t}$ since we are interested in the torque applied on the satellite. We have also neglected the contribution of non linear friction T_f.

$$P_{el}(T_{out}, \omega) = T_{out}^2 \frac{R}{k_t^2} + T_{out} \cdot \left(2B\frac{R}{k_t^2} + \frac{k_v}{k_t}\right)\omega + B\left(\frac{BR}{k_t^2} + \frac{k_v}{k_t}\right)\omega^2 \tag{3.6}$$

The power consumption P_{el} is a second degree polynomial in both T_{out} and ω.

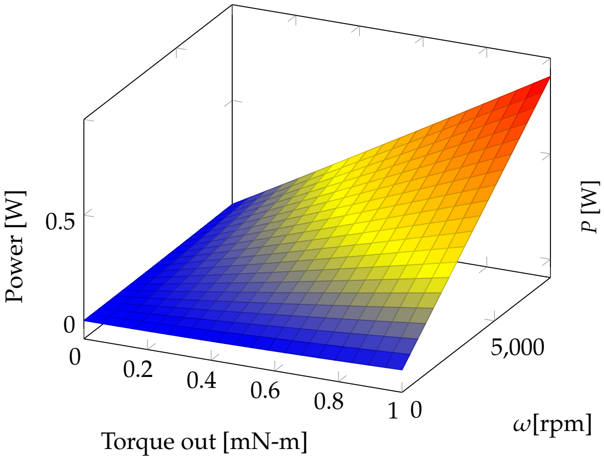


Figure 3.2: 3D rapresentation of Power necessary for a given torque at ω

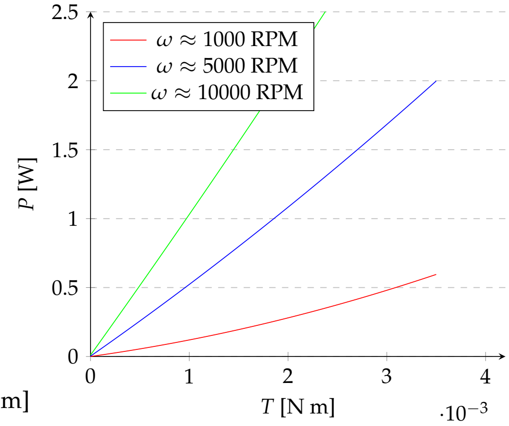


Figure 3.3: A visualization of P as a function of T and ω.

As clear from Fig. 3.2-3.3, power consumption for a given request varies considerably with ω.

3.2 Hardware characterization

The above model was quite simple and informative; however some parts of a real system are a bit harder to model. To increase the accuracy of the model and, most importantly, to ensure that we are not neglecting some important feature of the system, we will refine equation 3.6 with empirical data. Using lab hardware, we intend to measure power consumption under various ω, T_{out} conditions. Using data in the form $(P_{[W]}, \omega, T_{out})$ we will estimate the coefficients for model 3.6 with a best fit.

The tests are performed in atmosphere and with inexpensive hardware, unsuitable for flight; this choice was made to accommodate external constraints and to speed up the characterization campaign. The results, from the technological point of view, are pertinent only for order of magnitude estimation. From the modeling point of view however, the result ensures that the theoretical model is firmly grounded in reality. Using empirical data we are sure to accounts for the contribution of all the hardware elements (even those which would be hard to model analytically, like a micro controller or the Electronic Speed Controller).

The block diagram for the RW system is shown in 3.4. There are a few interesting differences with respect to the analytical model

- We utilize an Electronic Speed Controller (ESC) and avoid implementing low level electromechanical details.

- We measure total consumed power, which include the contribution of ESC and micro controller.

- The micro controller and ESC are integral part to have the RW work, but they would be hard to model.

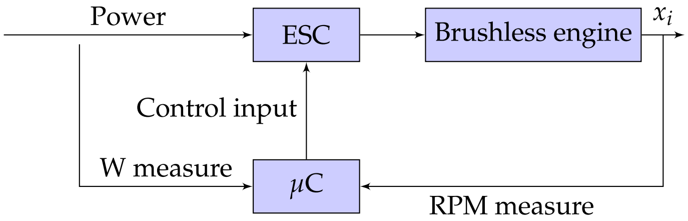


Figure 3.4: Schematic representation of the test RW model

We would like to measure consumed power given a command torque, at some initial state ω_0. To do so we need to implement a control law for which, in turn, we need to characterize the behavior of the subsystem brushless motor, ESC and micro controller. The test campaign has been structured in three phases:

1. Open loop steady-state characterization
2. Open loop dynamic characterization
3. Closed loop characterization
4. Real time characterization

It should be noted that the open loop dynamic is stable, not only due to the nature of the actuator, but also due to the control law embedded within the Electronic Speed Controller. The ESC introduces an additional level of uncertainty during dynamic characterization, as it might apply some unknown corrections to the behavior of the brushless engine.

3.2.1 Brief description of the hardware

The motor is an inexpensive brushless engine, rated for approximately 14.4 V and 9W, it was chosen for its size and because it was readily available. The ESC was chosen to match the requirement of the brushless engine, while the micro controller is an AVR ATMega328P (the common microprocessor found in the Arduino Uno board). There are only two sensors, one to measure the total current consumed by the actuator and one to detect the rotation of the rotor through a magnet embedded in it.

The moment of inertia of the rotor around the axis of rotation is estimated from its geometry at $I_w \approx 1.68 \times 10^{-5}$ kg m^2. The solid disk portion contributes to $I_d = 3.18 \times 10^{-6}$ kg m^2, while the external ring $I_r = 1.37 \times 10^{-5}$kg m^2 . Direct measurements of inertia would be hard to obtain.

The results of the tests are in units which are practical for measurement and implementation, such as RPM, mAmps, microseconds etc. Conversion to system level figures will be performed afterward. For example, in small satellites the torque is commonly expressed in milli Newton meter (mN m). Given that the angular accelerations are between ±200 RPM per second, torque output is around

$$T = 200 \cdot \frac{RPM}{s} \cdot \frac{2\pi}{60} \cdot 1.68 \times 10^{-5}\text{kg m}^2 \approx 0.35 \text{ mN m}$$

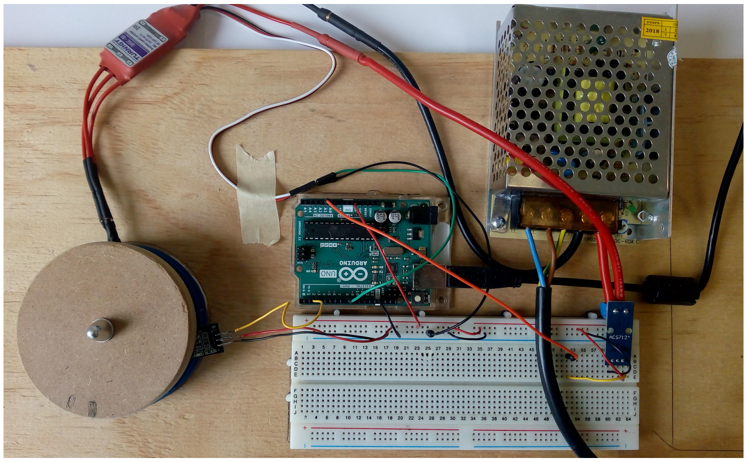


Figure 3.5: Picture of the hardware set up; brushless motor under the inertia disk, ESC (in red, connected to the motor),RPM sensor (connected to the motor by yellow, red and black wires), Arduino/controller (in green), Amp sensor (in blue) and power supply unit.

3.2.2 Open loop steady state characterization

The first model that needs to be developed is a map between the control input applied to the ESC (in this case, it is in PWM form) and the steady state angular velocity ω associated to it. Clearly ω has a dynamics, therefore we specify that we are interested in the *steady state* mapping; the speed which the rotor would *eventually* reach, given enough time.

$$f_1(u_{\text{PWM}}) = \omega_{t\infty} \quad \text{where} \quad \omega_{t\infty} \doteq \lim_{t \to +\infty} \omega(t) \tag{3.7}$$

Since we know that the open loop dynamic is stable, we can apply a set of commands (obtained by sampling the domain of the input u at regular intervals), allow the system to reach steady state by waiting a sufficiently long time, and then record the ω.

To ensure that $\omega_{t\infty}$ is well defined, for each command u_i two test are scheduled; in the first u_i is applied after u_{i-1} where $u_{i-1} < u_i$, in the second, the opposite is true ($u_{i-1} > u_i$). This assumes that the relationship between u and $\omega_{t\infty}$ is monotonic, which turned out to be the case. To limit the clutter in the plots, we introduce the characterization methodology with a sample of 9 tests, reported in table 3.2. The full set of 50 tests is used to derive the empirical laws and, when feasible, used to show general trends such as in figures 3.8 and 3.9.

Test n	Command PWM [%]	Measurements n	ω [RPM]	i [mAmp]
1	5	319	3776.2	96.6
2	8	429	5092.3	121.2
3	11	539	6374.1	154.3
4	14	419	7426.4	194.9
5	17	459	8236.4	237.4
6	14	419	7431.2	195.1
7	11	539	6409.2	155.7
8	8	439	5195.1	121.0
9	5	329	3919.5	96.2

Table 3.2: The summary of the outcome of a set of tests

The signals for RPM during the steady state phase are plotted as a function of time in Fig. 3.6; a visual check is sufficient to accept the validity of steady state condition. Readings for current during the same tests are shown in Fig. 3.7; here the signals are considerably more noisy in nature, but still appear to be stationary. The average value for each signal is taken and paired with the control input that generated it. The data for the complete test campaign (90 tests) are plotted in Fig. 3.8 and 3.9

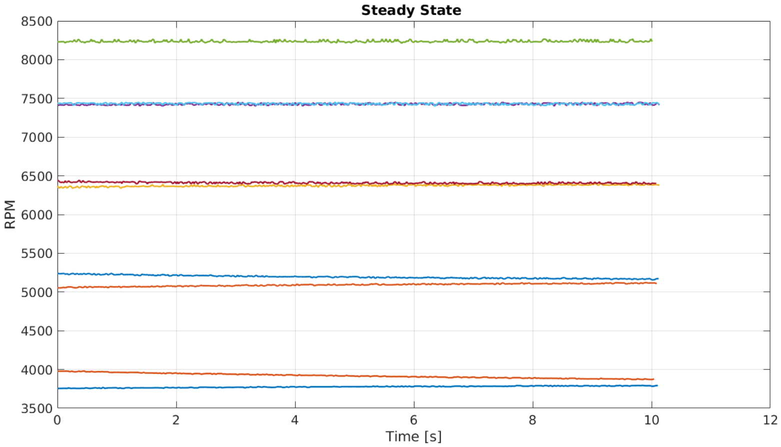


Figure 3.6: Time dependent RPM readings for different constant input signals (in PWM); these tests confirm that a constant PWM input leads to a constant RPM output, after a transitory period (not shown in figure).

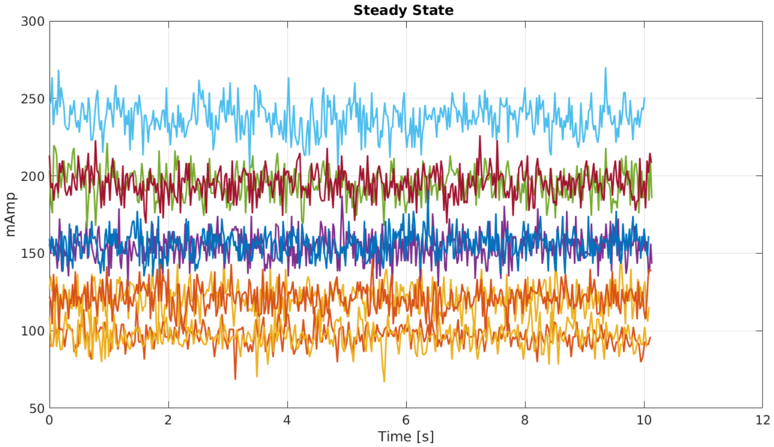


Figure 3.7: Time dependent mAmp readings for various constant input signals (in PWM); current reading show large deviation with respect to the mean, which appears to be in steady state.

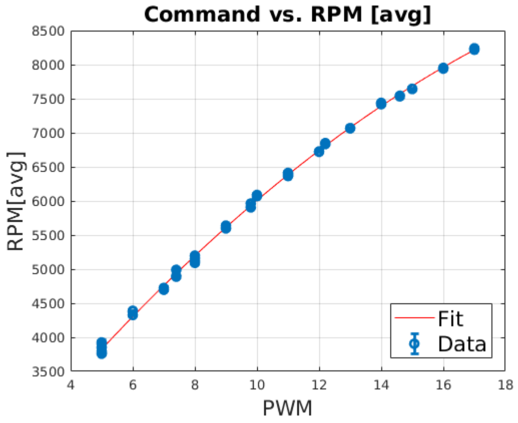


Figure 3.8: Control input (PWM [%]) Vs average steady state output (RPM)

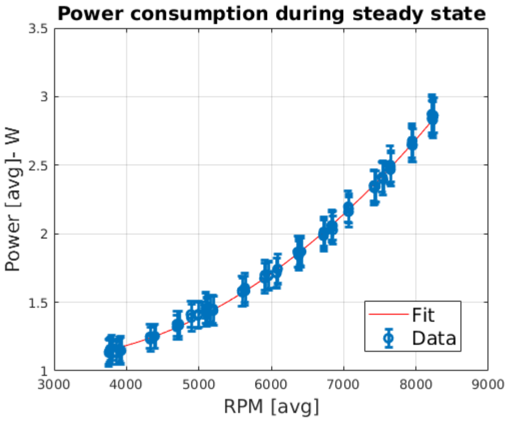


Figure 3.9: Steady state power consumption map against average data points

The best polynomial fit of second degree is (with 95% confidence bounds)

$$\omega_{[\mathrm{RPM}]} = a \cdot u_{[\%]}^2 + b \cdot u_{[\%]} + c \qquad \begin{cases} a & = & -10.09 & (& -10.84 & , & -9.333 &) \\ b & = & 587.9 & (& 571.5 & , & 604.2 &) \\ c & = & 1138 & (& 1058 & , & 1219 &) \end{cases} \tag{3.8}$$

With fit performances SSE: 1.8289e+05,$R^2 = 0.9990$

Power consumption in the absence of torque At steady state, there is no torque output[1]. Using the analytical model derived in the previous section (Eq. 3.6) we expect power consumption to be quadratic with ω. Using the average value for both the current signal and the RPM signal during each of the steady state tests we can obtain the plot 3.9. The second degree polynomial that best approximates the dataset is presented in 3.9.

$$P_{[W]} = a \cdot \omega^2_{[RPM]} + b \cdot \omega_{[RPM]} + c \quad \begin{cases} a &= 5.593 \times 10^{-8} & (& 5.35 \times 10^{-8} & , & 5.837 \times 10^{-8} &) \\ b &= -2.955 \times 10^{-4} & (& -3.247 \times 10^{-4} & , & -2.663 \times 10^{-4} &) \\ c &= 1.471 & (& 1.387 & , & 1.555 &) \end{cases} \tag{3.9}$$

Fit performance SSE:0.0038 $R^2 = 0.9986$

The large disparity in the order of magnitude of the coefficients is due to disparity in the units of ω and P; RPM range between 2000 and 8000 RPM while power consumption is between 1 and 3 Watts.

3.2.3 Open loop dynamic characterization

To characterize the dynamic response of the actuator, a series of different control steps is applied from different initial conditions. Data for both ω and current are recorded during the transitory. In Fig. 3.10 we report the ω signals for multiple tests. The time axis is set relative to the beginning of the test. We can observe that starting from different ω_0 we tend toward the same ω_{t_∞}. As we are interested in the dynamic part, the experiments are terminated before reaching the *final output*. The test performed are reported in table 3.3

[1] No useful torque applied on the satellite; the brushless engine is still overcoming friction

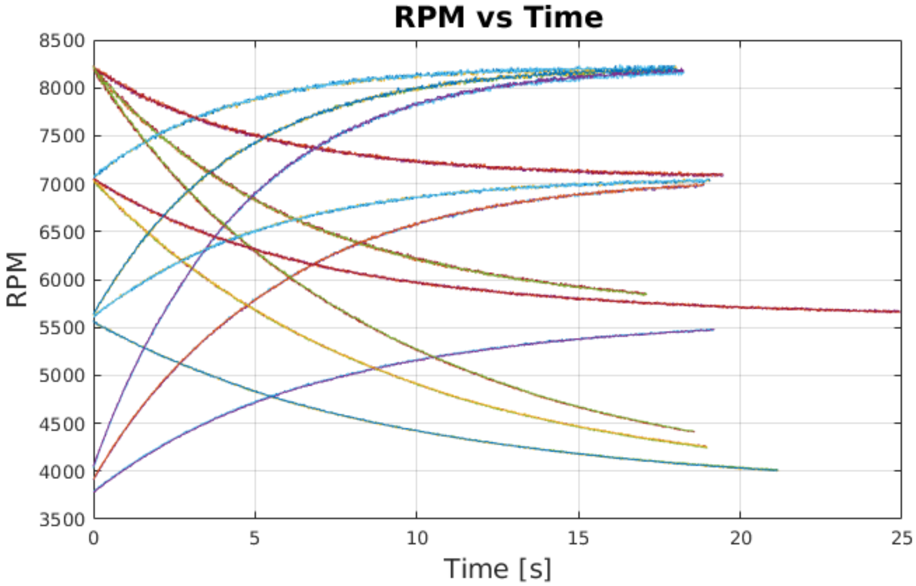


Figure 3.10: RPM vs Time signals for haviside step input of different amplitude (in PWM); the transitory response is shown and suggests that the motor dynamics can be approximately consdiered linear.

From the responses to input steps shown in Fig. 3.10 it is natural to model the response as

$$\omega(t) = \omega_{\text{target}} + \Delta\omega_0 \cdot e^{-k \cdot t} \qquad \Delta\omega_0 \doteq \omega_0 - \omega_{\text{target}} \tag{3.10}$$

We also observe that, for the same $|\Delta\omega_0|$, the dynamic response while increasing ω is quicker than that while decreasing it. This non symmetric behavior will need to be accounted for while implementing the control loop. Hence we need to predict the control stiffness k as a function of ω_0 and $\Delta\omega$.

We process every signal individually; the best fit for k given $\Delta\omega$ and ω_0 is reported in table3.3, along with 95% trust bounds.

Test	ω_0	ω_{target}	k	k$^-$	k+
N	RPM	RPM	[1/s]	[1/s]	[1/s]
1	3789.79	5611.81	0.14032	0.14012	0.14052
2	8232.71	5611.81	0.14967	0.14926	0.15007
3	7078.81	8216.29	0.24374	0.24289	0.2446
4	8232.71	7075.49	0.20464	0.20394	0.20534
5	5559.67	3825.25	0.10645	0.10632	0.10657
6	3916.45	7075.49	0.18294	0.18274	0.18314
7	7035.65	3825.25	0.10854	0.10844	0.10863
8	4040.95	8216.29	0.23117	0.23076	0.23159
9	8196.72	3825.25	0.11187	0.1117	0.11204
10	5622.19	7075.49	0.18878	0.1884	0.18915
11	7045.56	5611.81	0.13927	0.13898	0.13956
12	5641.22	8216.29	0.23813	0.2377	0.23856
13	3785.01	5611.81	0.14048	0.14025	0.1407
14	8228.2	5611.81	0.14713	0.14681	0.14746
15	7048.87	8216.29	0.2549	0.25395	0.25585
16	8214.68	7075.49	0.19543	0.19488	0.19598
17	5561.74	3825.25	0.10758	0.10748	0.10768
18	3909.3	7075.49	0.18265	0.18242	0.18287
19	7025.76	3825.25	0.10848	0.10841	0.10856
20	4034.43	8216.29	0.23094	0.2306	0.23129
21	8196.72	3825.25	0.11168	0.1115	0.11185
22	5607.48	7075.49	0.18861	0.18828	0.18894
23	7052.19	5611.81	0.14036	0.14006	0.14065
24	5630.63	8216.29	0.23952	0.23903	0.24001
25	3783.1	5611.81	0.13929	0.13911	0.13948
26	8196.72	5611.81	0.14666	0.14641	0.14692
27	7075.47	8216.29	0.24952	0.24858	0.25047
28	8196.72	7075.49	0.19545	0.19484	0.19606
29	5570	3825.25	0.10753	0.1074	0.10765
30	3917.47	7075.49	0.18309	0.18286	0.18332
31	7038.95	3825.25	0.1084	0.1083	0.10849
32	4049.68	8216.29	0.2319	0.23145	0.23235
33	8232.71	3825.25	0.11231	0.11211	0.11252
34	5613.77	7075.49	0.18908	0.18873	0.18943
35	7055.5	5611.81	0.14023	0.13995	0.14051
36	5636.98	8216.29	0.23974	0.23927	0.24022

Table 3.3: Best fits for the parameter k, with 95% upper and lower bounds

As k varies sensibly, we interpolate its value as a function of the starting ω_0 and the requested

$\Delta\omega$

$$k = a \cdot \Delta\omega_{[\mathrm{RPM}]} + b \cdot \omega_{0,[\mathrm{RPM}]} + c \quad \begin{cases} a & = & -3.042 \times 10^{-5} & (& -3.261 \times 10^{-5} & , & -2.824 \times 10^{-5} &) \\ b & = & 3.34 \times 10^{-5} & (& 2.983 \times 10^{-5} & , & 3.698 \times 10^{-5} &) \\ c & = & -0.0361 & (& -0.05851 & , & -0.01369 &) \end{cases} \tag{3.11}$$

Performance for the fit are SSE: 0.0033, $R^2 = 0.9624$

Results for the fit are presented in Fig. 3.11.

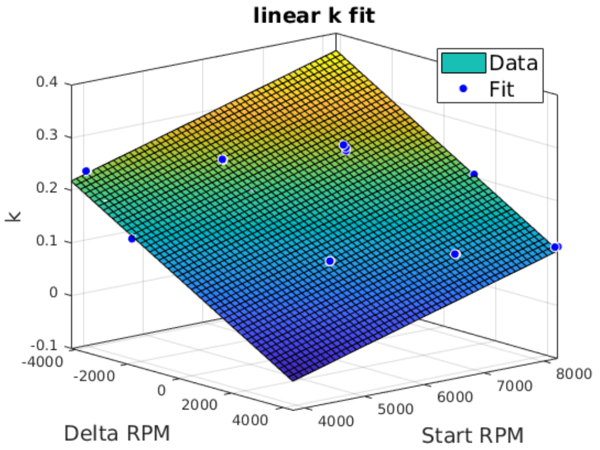


Figure 3.11: Fit for control stiffness parameter k

3.2.4 Close loop control

A simple closed control loop is implemented and quickly characterized by setting target angular accelerations (in RPM/s). Results, shown in Fig.3.12, provide some assurance on the consistency of the output; as the angular velocity increase linearly with time, its derivative and hence the torque are approximately constant. Fig. 3.13 shows a good agreement between the request and the actuator response, measured as the average angular acceleration for each signal.

The torque output from the RW is proportional to the angular acceleration. Using model 3.10, we can directly control the instantaneous value of the derivative as

$$\frac{\mathrm{d}\omega}{\mathrm{d}t}(t) = -\Delta\omega_0 k \cdot e^{-kt} \quad \lim_{t \to 0} \frac{\mathrm{d}\omega}{\mathrm{d}t}(t) = -\Delta\omega_0 k \tag{3.12}$$

We implement the following control scheme;

1. Read current RPM and desired ${}^{\mathrm{d}\omega}/_{\mathrm{d}t}$
2. Set a first approximation of ω_{target} using the average value for k

$$\omega_{\mathrm{target}} = {}^{\mathrm{d}\omega}/_{\mathrm{d}t} \cdot {}^{1}/_{k_{avg}} + \omega_0$$

3. Refine the value for k using the first approximation of $\Delta\omega$ according to 3.11

4. Compute a more precise ω_{target} with a better approximation of k and apply the control input necessary to reach it at steady state.

5. repeat to convergence

To prove that the iterative method converges, one can use Banach fixed point theorem. If one shows that the function $f : \Omega \to \Omega$ used in the iteration allows, under a suitable metric $|| \cdot ||$, for the existence of $q \in [0, 1)$ such that

$$||f(x) - f(y)|| \leq q \cdot ||x - y|| \tag{3.13}$$

then, there is a unique fixed point $x^\star$ such that $f(x^\star) = x^\star$, and every chain of iteration will lead to it ($\lim_{n\to\infty} f(x_0)^n \to x^\star \; \forall x_0$). In this case, f is a function that improves the estimate of the value of k, namely

$$f(k_0) = a \cdot \frac{T}{k_0} + b \cdot \omega_0 + c \quad = k_1 \quad \text{as per Eq. 3.11} \tag{3.14}$$

Using the standard *absolute value* norm, this propriety is verified whenever $a \cdot T < k_1 \cdot k_2$, which, accounting for the numerical values of the constants, ensures convergence for $k \in [0.14, 0.30]$.

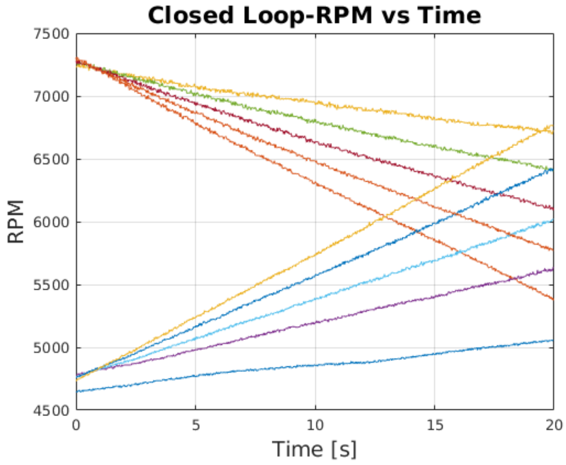


Figure 3.12: A set of output signals for various RPM/S requests, both increasing and decreasing

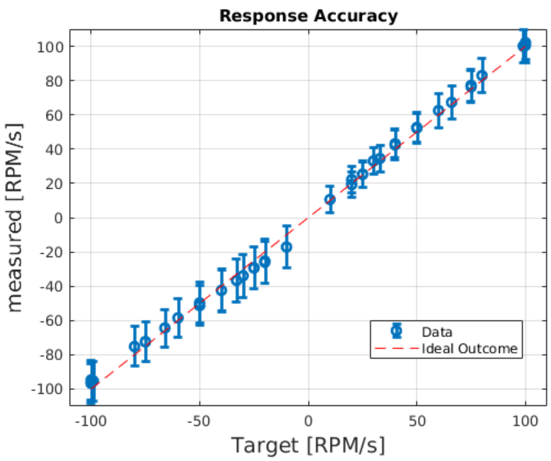


Figure 3.13: Request vs output; measured angular acceleration is averaged over a signal of approximately 20 seconds

Results for the full sets of 42 tests are reported in table 3.4.

Test	Target $\dot{\omega}$	Measured $\dot{\omega}$	Current	Dynamic Current
N	RPM/S	RPM/s	mAmp	mAmp
1	25	25.19	126.4	12.2
2	-25	-29.39	175.9	-0.5
3	50	52.61	140.8	17.2
4	-50	-51.31	163.7	-5.6
5	75	76.29	150.7	21.8
6	-75	-72.23	150.4	-11
7	100	100.44	163.2	28.4
8	-100	-96.7	136.3	-16.6
9	25	25.4	124.8	10.8
10	-25	-29.17	175.7	-0.9
11	50	52.23	139.6	16
12	-50	-51.09	162.6	-6.3
13	75	77.24	151.7	22.5
14	-75	-72.49	150.5	-10.7
15	100	100.97	162.3	27.2
16	-100	-96.25	134	-18.8
17	33	34.54	126.1	10.1
18	-33	-36.6	170.7	-3
19	66	67.29	144.4	17.3
20	-66	-64.36	152.4	-12
21	99	100.07	160.5	25.6
22	-99	-95.26	135.9	-17.2
23	20	19.3	121.3	8.3
24	-100	-94.72	134.6	-18.7
25	-20	-26	177.1	0.1
26	40	42.41	134.4	13.5
27	-40	-42.47	167.4	-4.3
28	60	62.47	144.4	18.3
29	-60	-58.51	156.4	-10
30	80	82.97	151.9	21.3
31	-80	-75.22	146.7	-12.9
32	100	102.25	161.7	26.2
33	10	10.6	119	7.8
34	-50	-49.69	162.8	-6.5
35	-10	-17.1	183.8	3.2
36	20	22.25	126.5	10.6
37	-20	-25.3	180.4	2.6
38	30	33.07	131.5	12.9
39	-30	-33.92	171.6	-3.3
40	40	43.14	135.3	14.1
41	-40	-42.46	167.5	-4.3
42	50	52.62	139	15.5

Table 3.4: Measures from closed loop actuation

Power consumption model

We test the analytical model 3.6 in the simplified version below

$$P_{el}(T_{out}, \omega) = A \cdot T_{out}^2 + T_{out} \cdot B \cdot \omega + C \cdot \omega^2$$

against the set of all data points collected for the closed loop characterization. The points are in the form $(\frac{\mathrm{d}\omega}{\mathrm{d}t}, \omega_{[\mathrm{RPM}]}, i_{[\mathrm{mAmp}]})$ so the fit is more like

$$i_{[\mathrm{mAmp}]}\left(\frac{\mathrm{d}\omega}{\mathrm{d}t}, \omega\right) = A^* \cdot \left(I_w \frac{\mathrm{d}\omega}{\mathrm{d}t}\right)^2 + \left(I_w \frac{\mathrm{d}\omega}{\mathrm{d}t}\right) \cdot B^* \cdot \omega_{[\mathrm{RPM}]} + C^* \cdot \omega_{[\mathrm{RPM}]}^2$$

Fitting for the values $A' = A^* \cdot I_w^2, B' = B^* \cdot I_w, C$ we obtain a poorly predictive model

$$\begin{array}{ccccccc} A' & = & 0.0003702 & (& 0.0003224 & , & 0.0004181 &) \\ B' & = & 5.988 \times 10^{-5} & (& 5.943 \times 10^{-5} & , & 6.034 \times 10^{-5} &) \\ C & = & 4.056 \times 10^{-6} & (& -1.792 \times 10^{-6} & , & 5.974 \times 10^{-6} &) \end{array} \tag{3.15}$$

With performance SSE: 7.7604e+06, $R^2 = 0.5512$, which is not great. A considerably better model can be obtained by adding a constant; the equation becomes

$$i_{[\mathrm{mAmp}]}\left(\frac{\mathrm{d}\omega}{\mathrm{d}t}, \omega\right) = A_2 \cdot \left(\frac{\mathrm{d}\omega}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}\omega}{\mathrm{d}t}\right) \cdot B_2 \cdot \omega_{[\mathrm{RPM}]} + C_2 \cdot \omega_{[\mathrm{RPM}]}^2 + D_2$$

$$\begin{array}{ccccccc} A_2 & = & -3.023 \times 10^{-4} & (-3.324 \times 10^{-4} & , & -2.721 \times 10^{-4}) \\ B_2 & = & 3.727 \times 10^{-5} & (3.694 \times 10^{-5} & , & 3.761 \times 10^{-5}) \\ C_2 & = & 2.616 \times 10^{-6} & (2.603 \times 10^{-6} & , & 2.629 \times 10^{-6}) \\ D_2 & = & 56.86 & (56.39 & , & 57.34) \end{array} \tag{3.16}$$

SSE: 2.9656E+06, R^2: 0.8285. We will refer to this model as the Modified Analytical.

A still better approximation can be found by separating the dynamic contribution from the static one. We define a dynamic current as the difference between measured value and the expected one at steady state

$$i_{\mathrm{dyn}}\left(\frac{\mathrm{d}\omega}{\mathrm{d}t}, \omega\right) \doteq i_{\text{tot, measured}}\left(\frac{\mathrm{d}\omega}{\mathrm{d}t}, \omega\right) - i_{\text{Steady State}, T=0}(\omega) \tag{3.17}$$

Now we have that the steady state contribution accounts for ω and the dynamic contribution depends only on T_{req}. Having applied an approximately constant torque, we can use the average of the signals to obtain a model of the dynamic contribution of power (over the range of ω)

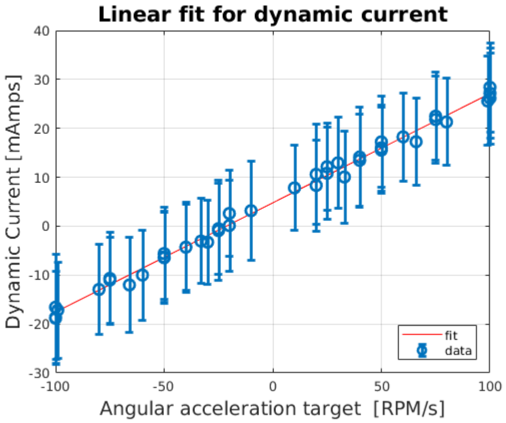


Figure 3.14: Linear fit for dynamic current as a function of requested angular acceleration

Data can be found in table 3.4.
With current values, best interpolation for i_{dyn} is

$$i_{dyn}\left(\frac{\mathrm{d}\omega}{\mathrm{d}t}\right) = a \cdot \frac{\mathrm{d}\omega}{\mathrm{d}t}_{\,[\mathrm{RPM/s}]} + b \qquad \begin{cases} a &= & 0.2233 & (0.2177 & , & 0.2289) \\ b &= & 4.787 & (4.443 & , & 5.131) \end{cases} \tag{3.18}$$

Performance for the fit are sse: 48.7010, $R^2 = 0.9940$

We compare the prediction power of both models against the complete punctual dataset obtained during the close loop characterization. We will call the first the analytical model

$$i_1\left(\frac{\mathrm{d}\omega}{\mathrm{d}t}, \omega\right)_{[\mathrm{mAmp}]} = A' \cdot \frac{\mathrm{d}\omega}{\mathrm{d}t}^{2}_{\,[\mathrm{RPM/s}]} + \frac{\mathrm{d}\omega}{\mathrm{d}t}_{\,[\mathrm{RPM/s}]} \cdot B' \cdot \omega_{[\mathrm{RPM}]} + C \cdot \omega^2_{[\mathrm{RPM}]} \tag{3.19}$$

and the second the empirical one

$$i_2\left(\frac{\mathrm{d}\omega}{\mathrm{d}t}, \omega\right)_{[\mathrm{mAmp}]} = \left(a_{\mathrm{dyn}} \cdot \frac{\mathrm{d}\omega}{\mathrm{d}t}_{\,[\mathrm{RPM/s}]} + b_{\mathrm{dyn}}\right) + \frac{1000}{12}\left(a_{\mathrm{ss}} \cdot \omega^2_{[\mathrm{RPM}]} + b_{\mathrm{ss}} \cdot \omega_{[\mathrm{RPM}]} + c_{\mathrm{ss}}\right) \tag{3.20}$$

Results for the two model in predicting mAmp values given ω and T are reported in table 3.5

Model	SSE	R^2
Analytical	1.2×10^9	0.4936
Empirical	2.9×10^6	0.82634

Table 3.5: Prediction current of the two model

Summary of the models

From now on, we will consider both models.For ease of reference, they are reported here with the proper coefficients and as a function of power and with multiple possibilities for units of output and input

	Output	Input	Equation	a	b	c	d
AA	[W]	[RPM/s]	$a \cdot \dot{\omega}^2 + \dot{\omega} \cdot b \cdot \omega + c \cdot \omega^2$	5.644×10^{-6}	1.761×10^{-6}	3.292×10^{-10}	/
A0	[W]	[RPM/s]	$a \cdot \dot{\omega}^2 + \dot{\omega} \cdot b \cdot \omega + c \cdot \omega^2$	4.442×10^{-6}	7.185×10^{-7}	4.867×10^{-8}	/
M0	[W]	[RPM/s]	$a\dot{\omega}^2 + \dot{\omega}b\omega + c\omega^2 + d$	-3.627×10^{-6}	4.47×10^{-7}	3.139×10^{-8}	0.6823
E0	[W]	[RPM/s]	$a \cdot \dot{\omega} + b \cdot \omega^2 + c \cdot \omega + d$	2.679×10^{-3}	5.593×10^{-8}	-2.954×10^{-4}	1.528
A1	[mAmp]	[RPM/s]	$a \cdot \dot{\omega}^2 + \dot{\omega} \cdot b \cdot \omega + c \cdot \omega^2$	3.702×10^{-4}	5.988×10^{-5}	4.056×10^{-6}	/
M1	[mAmp]	[RPM/s]	$a\dot{\omega}^2 + \dot{\omega}b\omega + c\omega^2 + d$	-3.023×10^{-4}	3.727×10^{-5}	2.616×10^{-6}	56.86
E1	[mAmp]	[RPM/s]	$a \cdot \dot{\omega} + b \cdot \omega^2 + c \cdot \omega + d$	0.2233	4.661×10^{-6}	-2.462×10^{-2}	127.37
A2	[W]	[mNm]	$a \cdot T^2 + T \cdot b \cdot \omega + c \cdot \omega^2$	1.435	4.084×10^{-4}	4.867×10^{-8}	/
E2	[W]	[mNm]	$a \cdot T + b \cdot \omega^2 + c \cdot \omega + d$	1.523	5.593×10^{-8}	-2.954×10^{-4}	1.528

Table 3.6: Coefficients obtained by fitting the empirical data for various units of inputs and outputs.

3.2.5 Notes on real time behavior

In order to implement any Real Time (RT) control system, we need to have an estimate of the maximum time required to complete each phase in the control loop. The most essential control loop has three steps; reading sensors, estimating the current state of the system, compute and apply the control input. Times for each of these functions have been tabulated in a variety of different conditions; in table 3.7 we present the results:

Phase	Avg	Max
	ms	ms
Get Measure	72.3	89.9
Estimate State	0.47	0.492
Compute Control	0.41	0.43

Table 3.7: Response times of each of the phases in the control loop.

Clearly, the most time consuming phase is data acquisition. The reason is easy to understand; since the rotor has a single magnet on it, RPM can be measured only with a full turn. Given that the lowest speed used is around 3000 rpm, to measure a single revolution it can take up to 20 ms. Furthermore, as a revolution is measured as the time between two consecutive rising edges observed by the hall sensor, it might take as much as 40 ms. Since three measures are taken each time and averaged out a *single* RPM measure might take as much as 120 ms.

With these data, the quickest full loop hard real time control is at around 8 Hz, much of which is spent waiting for the hall sensor to detect the magnet, which is not a good use of the resources. However, since state estimation is rather cheap (and accurate), we could allow measurement to be soft real time and preempt it in case of missed deadline.

The addition of more markers at fixed angular interval would decrease wait time. An attempt was made with 8 magnets, hoping to achieve a maximum time of around 12 ms and a 100 Hz control loop. However, by reducing the distance between the markers the magnetic field became too homogeneous to trigger the hall sensor, thus preventing any reading. As we had no real requirement on the real time behavior, we settled for the rotor with the single marker.

3.3 Cost functions

With a single actuator, we do not have degrees of freedom for optimization; requested torque must be met by the actuator so $T_{\mathrm{RW}}(t) = T(t)$, and considering initial condition $\omega(t=0)$, both $\omega(t)$ and the consumed power are determined. The models developed so far could be useful for optimization during the design of the system, but not to improve efficiency during operations. To pursue operation optimization, we will consider the case with multiple RWs capable of producing torque in the same direction.

The objective is to minimize power consumption while producing the requested torque T_0. The intuitive choice for the cost function is to use power consumption, in our case either Eq. 3.19 or 3.20. However, both models show that the internal state of a RW, its rotational speed ω, contributes significantly to the power consumption; therefore the general model we will considered is $\mathcal{C}(T,\omega)$. Given the vector of angular velocities for each RW in the cluster $(\omega_0^1, \omega_0^2 \ldots, \omega_0^n)$, which acts as an initial condition, we can pursue a torque allocation that minimizes total cost. We solve an optimization under equality constraint

$$\text{find } \vec{T} = \begin{pmatrix} T_1 \\ T_2 \\ \ldots \\ T_n \end{pmatrix} \text{ such that } \begin{cases} \sum_{i=1}^{n} \mathcal{C}(T_i, \omega_0^i) & = & \min\limits_{\vec{T}} \sum_{i=1}^{n} \mathcal{C}(T_i, \omega_0^i) \\ \quad \text{and} & & \\ \sum_{i=1}^{n} T_i & = & T_0 \end{cases} \tag{3.21}$$

After the solution is found, we implement it and repeat the process for the next request T_1. However, we need to update the initial conditions $(\omega^1, \omega^2 \ldots, \omega^n)$, integrating the torque allocation used in the previous step. Hence, due to the internal dynamics of the RWs, the solution of the problem at any given step affects the solution of all later instants, and thus, minimizing overall power consumption becomes a dynamic problem.

To simplify the problem, we can choose to ignore or work around this dependency; we will explore statistical ideas to do so in section 3.3.1. However, we might understandably be uncomfortable with this approach. After all, similar problems are often found in control theory, where we steer the system toward a target state over the course of several time steps, aiming to minimize some cost function defined over a time horizon. This approach however requires prior knowledge of torque request as a function of time, which is a rather taxing assumption.

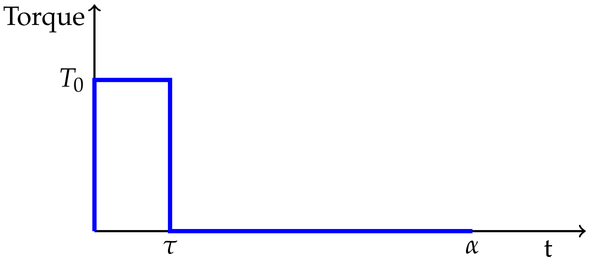


Figure 3.15: Depiction of the pilot torque signal, parameteric with τ and α

Consider a torque request signal as shown in Fig 3.15. The shape is suggested by the typical usage of a RW; torque is requested in short bursts followed by relatively longer periods of coasting.

At time $t = 0$, we are requested a torque level T_0; this is to be maintained up to a time τ and then dropped until the time horizon, instant α. If we knew the parameters α, τ we could minimize average power consumption up to the time horizon. Since we do not know it we could hypothesize it, but clearly different guesses will lead to a different solution. This is shown in Fig 3.16, where two allocation points respond to the same instantaneous torque request but minimize different *expected* power consumption.

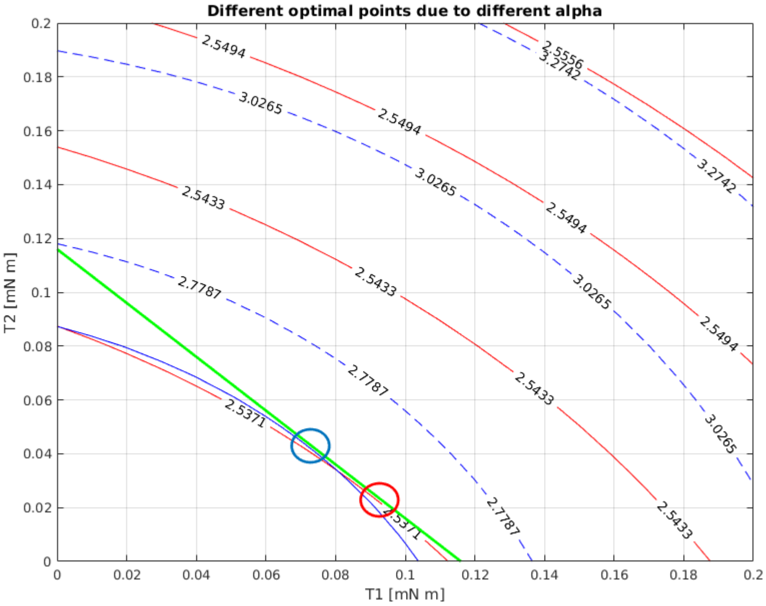


Figure 3.16: Using different values of α the optimal allocation among two RW acting in the same direction to supply ≈ 0.12 mN m changes. The green line shows all the combinations of T1 and T2 able to meet the requested output torque. The red and bluecurved lines show equi-cost lines for two different values of α. As we would expect, the optimal regions, circled in red and blue change as well, depending on the choice of α / cost model. (Graphically we can recognize the constrained stationary points as the points in which the equi-cost lines are tangent to the constraint line)

A dual of the above argument would be to have some statistical knowledge of the shape of the typical request profile and therefore have some estimation of α, τ. Then we can define a parametric cost function for the expected power consumption averaged over the time horizon:

$$C(T_0, \omega_0)|_{\alpha,\tau} = \frac{1}{\alpha}\int_0^{\alpha} P(T(t), \omega(t))\mathrm{d}t = \frac{1}{\alpha}\int_0^{\tau} P\left(T_0, \omega_0 + \frac{T}{I_w}\cdot t\right)\mathrm{d}t + \frac{1}{\alpha}\int_{\tau}^{\alpha} P\left(0, \omega_0 + \frac{T}{I_w}\cdot \tau\right)\mathrm{d}t \tag{3.22}$$

The unit of cost is now an expected power [W]. Intuitively, if $\alpha \gg \tau$ we are concerned with the long term effect of instantaneous request, while $\alpha \approx \tau$ express deeper interest in the immediate repercussion on power demand. Figures 3.17 shows the effect of choosing one or the other; changing the cost function changes the objective of the optimization and therefore its outcome.

It is important to remark that while power consumption is something we can measure directly, the cost function is something we define artificially. It is an heuristic that shapes the optimization; a simplification we accept given that we can not optimize for power consumption unless we know $T(t)$ exactly. How well it approximates our goal can only be assessed a posteriori.
Its definition is therefore a responsibility of the system engineer. The aim of this section is to highlight the fact that, contrary to the case of a single actuator, the definition of cost within a cluster is not straight forward. Nonetheless, once the cost function is defined we can faithfully implement it with various algorithms.

In figure 3.17 we present the impact that the choice of parameter α has on the cost function, in red derived from the empirical model (Eq. 3.20) while in blue from the analytical one (Eq. 3.19)

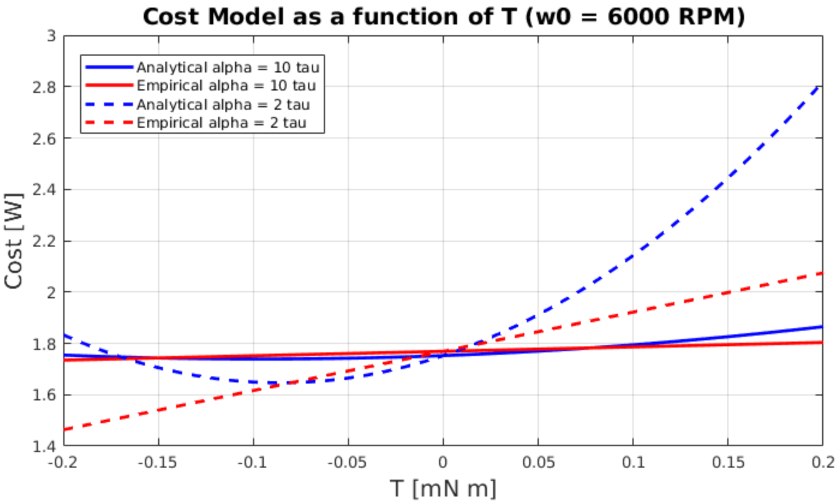


Figure 3.17: Cost functions according to Eq. 3.22 for the Analytical and Empirical model. Two different values of α are considered ($\tau = 100$ms).

3.3.1 Statistical approach

To avoid the dynamic problem 3.21 we can work around the internal variable ω to decouple the various time steps. The idea is to consider the effect of ω globally rather than punctually; the power consumed during the whole mission is assumed to be governed by the statistical properties of the signal ω and not by the specific profile $\omega(t)$. We hypothesize that we can use a function

$$\mathcal{P}(T) \text{ such that } \int_0^{t_{\text{end}}} \mathcal{P}(T(t))\mathrm{d}t = \int_0^{t_{\text{end}}} P(T(t), \omega(t))\mathrm{d}t \tag{3.23}$$

With knowledge of $\mathcal{P}(T)$ we can use it as the cost function, which no longer depends on ω. Generally, it is easy to prove that such function exits, but this is clearly not sufficient. We need to have a way to define it either analytically, numerically from simulations or from information gathered in similar previous missions. If $P(T, \omega)$ is separable in T and ω, we can easily define $\mathcal{P}$. Note that the empirical model 3.20 is separable

$$\int_0^{t_{\text{end}}} P(T(t), \omega(t))\mathrm{d}t = \int_0^{t_{\text{end}}} f_1(T(t)) + f_2(\omega(t))\mathrm{d}t = \int_0^{t_{\text{end}}} f_1(T(t)) + \frac{1}{t_{\text{end}}}\left(\int_0^{t_{\text{end}}} f_2(\omega(t))\mathrm{d}t\right)\mathrm{d}t \tag{3.24}$$

And we can write

$$\mathcal{P}(T) = f_1(T) + f_2|_{\text{avg}} \tag{3.25}$$

The definition of $f_2|_{\text{avg}}$ is still strictly dependent on the mission, but it has a clear meaning and we can estimate it. A value for $f_2|_{\text{avg}}$ can be obtained by assuming a probability distribution function for the angular velocities of the RW (pdf(ω))

$$f_2|_{\text{avg}} = \frac{1}{t_{\text{end}}}\int_0^{t_{\text{end}}} f_2(\omega(t))\mathrm{d}t = \int_{\omega_{\text{min}}}^{\omega_{\text{min}}} \omega \cdot \text{pdf}(\omega)\mathrm{d}\omega \tag{3.26}$$

The integral in 3.26 should be interpreted as a Lebesgue integral, the formal definition and use of which is outside the scope of this work. For our purpose it is enough to recall that it extends the standard notion of integral and thus when the Riemann integral is defined, the two have the same value. For simple probability distribution functions, we can just use the Riemann integral.

Assuming that ω is randomly distributed over the mission, we can compute a value for $\mathcal{P}$ (as based on the empirical model 3.20)

$$\mathcal{P}(T) = aT + \left(\frac{b\omega_{\text{max}}^2}{3} + \frac{c\omega_{\text{max}}}{2} + d\right) \tag{3.27}$$

Numerical coefficients for Eq. 3.27 are reported in table 3.8.

Non-separable $P(T, \omega)$ In this case, we require a stronger assumption to justify a similar result. We state that, at any time during the mission, the angular velocity of a RW is $\omega(t)$ independent of instantaneous torque request $T(t)$. This is to say that, when requesting an amount of torque, the probability that the state of the reaction wheel is $\omega = x$ depends only on its probability distribution. This is clearly a big simplification, but leads to a very general definition of $\mathcal{P}$

$$\mathcal{P}(T_{out}) = \int_{\omega_{min}}^{\omega_{max}} P_{el}(T_{out}, \omega) \cdot \text{pdf}(\omega)\,\mathrm{d}\omega \tag{3.28}$$

Assuming a probability distribution function on the domain $\omega = [0, \omega_{max}]$, we can find the expected power requirement for a given torque T_{out}. In figure 3.18 we show a constant probability distribution and a linear one. We will use them as examples to analytically derive the respective $\mathcal{P}$

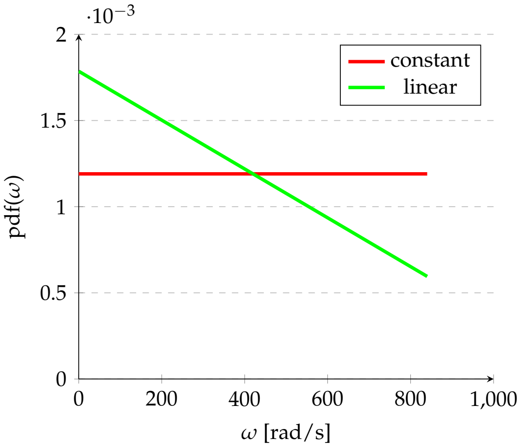


Figure 3.18: Different probability density function.

Assuming equi-probability for all values of ω, we obtain

$$\mathcal{P}_1(T_{out}) = \frac{1}{\omega_{max}} \cdot \int_0^{\omega_{max}} P_{el}(T_{out}, \omega) \mathrm{d}\omega$$

Thus integrating up to saturation speed ω_{max} according to the analytical model we obtain: [2]

$$\mathcal{P}_1(T_{out}) = \frac{R}{k_t^2} \cdot T_{out}^2 + \left(B \frac{R}{k_t^2} + \frac{k_v}{2k_t} \right) w_{max} \cdot T_{out} + w_{max}^2 \frac{B}{3} \left(\frac{BR}{k_t^2} + \frac{k_v}{k_t} \right)$$

	Output	Input	Equation	a	b	c
A2	[W]	[RPM/s]	$a \cdot \dot{\omega}^2 + \dot{\omega} \cdot b + c$	4.442×10^{-6}	3.233×10^{-3}	1.314
E2	[W]	[RPM/s]	$a \cdot \dot{\omega} + b$	2.679×10^{-3}	1.70	

Table 3.8: Coefficients for the expected power consumption based on analytical and empirical models.

Similarly, we could chose a different pdf, such as the one shown in red in Fig. 3.18, which changes linearly from a starting value p_0 at $\omega = 0$ to a final value p_1 at $\omega = \omega_m$

$$\text{pdf}(\omega) \doteq \frac{1}{\omega_m} \frac{2}{p_0 + p_1} \left(p_0 + \frac{(p_1 - p_0)}{\omega_m} \omega \right) \tag{3.29}$$

Solving the integral in Eq. 3.28 we obtain obtain ($\Delta p \doteq p_1 - p_0$)

$$\mathcal{P}(T) = \left(\frac{2}{p_1 + p_0} \right) \cdot \left\{ \left[\left(\frac{\Delta p}{2} + p_0 \right) \frac{R}{k_t^2} \right] \cdot T^2 + \left[\left(\frac{\Delta p}{3} + \frac{p_0}{2} \right) \left(2B \frac{R}{k_t^2} + \frac{k_v}{k_t} \right) \omega_m \right] \cdot T + \right.$$
$$\left. + \left(\frac{\Delta p}{4} + \frac{p_0}{3} \right) \left(\frac{B^2 R}{k_t^2} + \frac{k_v}{k_t} \right) \omega_m^2 \right\}$$

[2] For more details, see section 3.4.8

Since we have statistically eliminated the dependence on ω, we have a second degree polynomial in the requested torque T. In Fig. 3.20, we show the expected power requirement for a given torque using only one RW.

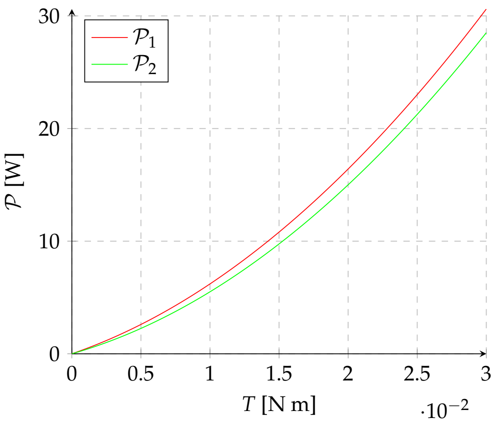


Figure 3.19: Large scale picture of expected power consumption $\mathcal{P}$ using different pdf.

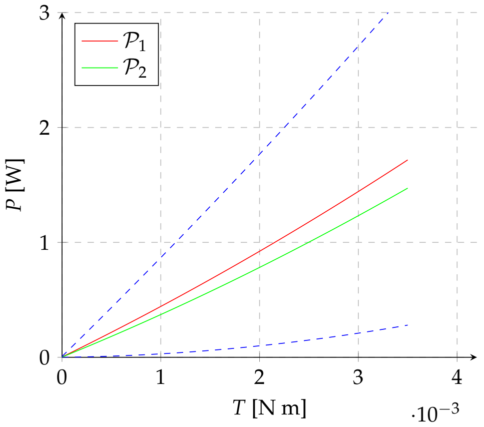


Figure 3.20: Small scale picture of expected power consumption $\mathcal{P}$ using different pdf.

Note that:

- Fig. 3.19 shows the trend of the power requirement for larger torque demands. Expected power is clearly non linear and there is much to be gained by using more actuators in parallel. For small torques however, as shown in Fig. 3.20, the linear approximation is rather good (at least under the hypothesis of equi ω distribution)

- Changing the probability distribution changes the expected cost. In green there is a linear pdf for ω, while in red there is the constant pdf for ω

- The dashed blue lines in Fig. **??** represent instantaneous power consumption at maximum and minimum ω. While we have used pdfs to filter out the dependency from ω, instantaneous power consumption still depends on it; the dotted blue lines are shown to provide context to the difference between pdfs. Clearly, regardless of the pdf, any $\mathcal{P}$ will be between the dotted blue lines.

3.4 Allocation strategies comparison

In this section;

- First we want to provide results to support the idea that more actuators can improve efficiency, if used correctly. We compare two algorithms which have similar properties, static allocation and the proposed method. For $n = 2$ we also include the analytical solution, which is helpful to gauge its complexity and to verify the optima of decentralized techniques.

- Once we have provided some proof that the benefits of decentralized frameworks might be worth the effort, we compare Dual Ascent with our method. The idea is to characterize the numerical properties of both algorithms, to show when they converge and how quickly when implemented in discrete time.

A note on the cost function used; initially we noticed the difference between the analytical cost function and the empirical one. More importantly, due to the dynamic nature of the problem, we have explained why cost is a choice and not something determined by the problem itself. Then we introduced multiple possible choices to deal with ω, such as hypothesizing a pilot signal or statistical approaches.
Performing all tests with all possible cost functions would be time consuming and would add little value to the exposition. Therefore we will strategically choose which cost function to use when. We consider the following:

- The analytical cost function, although a less accurate model, is of more general interest. Values for different brushless motors can be directly plugged in for first order estimate or sanity check. Furthermore, it is convex with torque, so Dual Ascent is assured to converge and we can properly compare the two decentralized methods.

- The empirical model and the modified analytical are more accurate, but perhaps of little general interest. Furthermore, they are both convex and thus can not be used to evaluate DA performances.

- Pilot signal models (based on either power consumption models) have a quite complex analytical formulation and are based on a choice of the parameter α and τ.

- Statistical method (based on either) require the assumption of a pdf but simplify the model considerably.

We choose to use the analytical method (Eq. 3.19) for almost all tests/comparisons. Notice that to account for different possible initial condition ω_0, we can think to run a large number of simulations, each time choosing a random initial condition ω_0. Then we take the average as the *true* value. This Monte Carlo approach will approximate the results we would find using the statistical method (assuming the same pdf is used in both cases) while retaining the effect of rotor dynamic and without excessive analytical complications. Therefore, when possible, we will speed up tests by using the statistical method directly instead of multiple experiment with randomly selected initial conditions.

3.4.1 Preliminary results with pairs of agents

To support the use of a cluster beyond the argument of reliability, we show how the efficiency profile changes with the number of agents. We initially present the options to control the cluster under the simplifying assumption of using only two agents.

A useful concept to compare allocation strategies is efficiency, and how it changes with request T and the state ω. Typically efficiency is an a-dimensional value; for an electric engine, the ratio between the mechanical power it outputs and the electrical power it requires. In the case of a RW however, we are more interested in the torque output, rather than the power output. Therefore we redefine efficiency as

$$\varepsilon^* \doteq \frac{T_{\text{out}}}{P_{el}} = \frac{J\frac{\mathrm{d}\omega}{\mathrm{d}t}}{i \cdot V} \tag{3.30}$$

Figure 3.22 highlights how increasing torque generally decreases efficiency. Unless we use a statistical estimation of power, ε^* will be a function of the rotor angular velocity (as well as a function of input power). Figure 3.21 shows a logarithmic plot of $\varepsilon^*(T, \omega)$

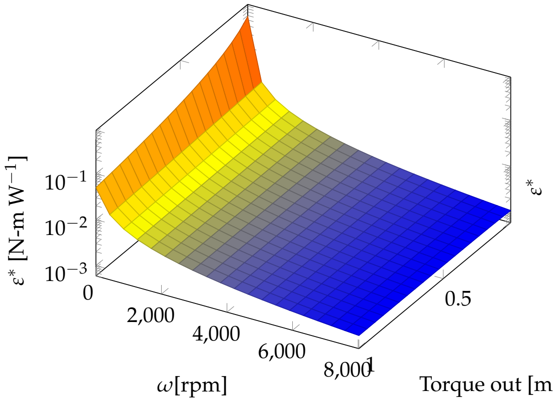


Figure 3.21: A more useful visualization of ε^* as a function of P_{el} and ω

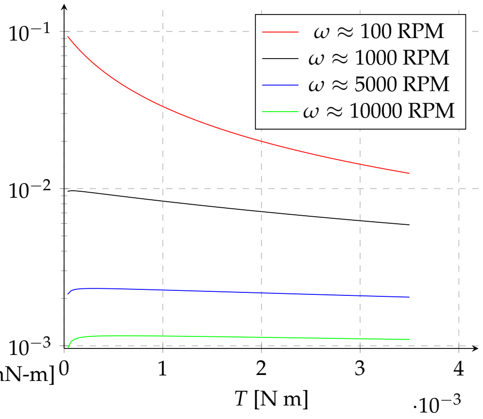


Figure 3.22: Efficency at different ω

We consider two main ways to reliably coordinate the cluster:

1. We observe that power consumption is super linear (as shown in Fig. 3.22) with torque. This means that efficiency increases for smaller outputs, so it is convenient to split request among as many agents as possible. Hence we can split the request equally among the n actuators, without regard for the individual internal state. This method is very simple to understand and implement and does not rely on communication[3]. As it does not account for the internal state dynamics of the agent, we will refer to it as Static Allocation (SA).

2. When the disparity in ω between two RWs is high, we would like to allocate more torque to the RW with lower ω. The decentralized approach can be used to this end, and in general to perform a proper optimization.

For the initial simple cases, we can consider a centralized allocation as a benchmark to set the optimum.

3.4.2 Static allocation

The rule we impose to translate the torque requested to the cluster T_{req} in the command for each RW is $T_{cmd,i} = \frac{T_{\text{req}}}{n}$. If expected power consumption is super linear, this improves efficiency. In

[3] It is still not very reliable as the central node, which allocates the torque to the others, needs to know exactly how many node there are. We assume that this can be measured independently and adjusted as the need arises.

other words, $T_1 < T_2 \rightarrow \mathcal{P}(T_1) < \mathcal{P}(T_2)$. Moreover, assuming that all RW are similar both in terms of electromechanical constants$(B, L, R, kv, kt, ...)$ and current state ω_i, the equal division strategy leads to the global minima. Then, the expected power consumption for the cluster can be expressed analytically as

$$\mathcal{P}(T_{req})|_n = \int_0^{\omega_{max}} n \cdot P\left(\frac{T_{req}}{n}, \omega\right) \cdot \mathrm{pdf}(\omega)\, \mathrm{d}\omega = n \cdot \mathcal{P}\left(\frac{T_{req}}{n}\right) \tag{3.31}$$

In Fig. 3.23 and 3.24 the expected power (assuming uniform probability) is plotted. The detail 3.24 shows marginal improvement for small clusters application.

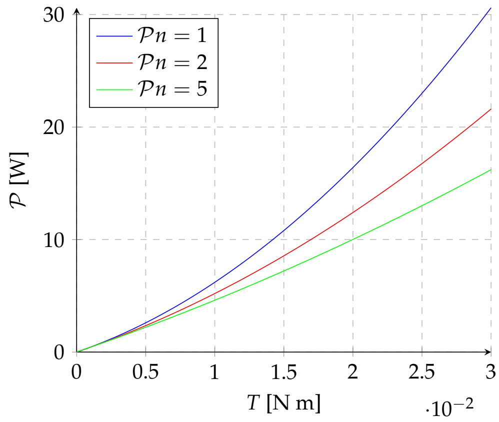


Figure 3.23: Large torque request (small sat) expected power for n=1,2,5.

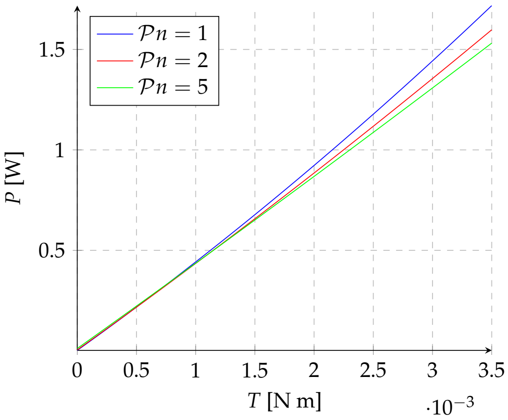


Figure 3.24: Detail of expected power for n=1,2,5

However, since we can not expect that all RWs will always maintain the same ω due to different use or because they have been produced differently (and therefore the same torque production needs different increments in ω) a more accurate definition of the expected power would be

$$\mathcal{P}(T_{req})|_n = \int_0^{\omega_{max}} \int_0^{\omega_{max}} \cdots \int_0^{\omega_{max}} \sum_i^n \left[P_{el}\left(\frac{T_{req}}{n}, \omega\right)\right] \cdot fq(\omega_1, \omega_2, ..., \omega_n)\, \mathrm{d}\omega_1 \mathrm{d}\omega_2 \ldots \mathrm{d}\omega_n \tag{3.32}$$

However, adding an assumption of independence between ω in the various RW, under the uniform probability distribution the result is the same.

$$\mathrm{pdf}(\omega_1, \omega_2, ..., \omega_n) = \mathrm{pdf}(\omega_1) \cdot \mathrm{pdf}(\omega_2) \cdot ... \cdot \mathrm{pdf}(\omega_n)$$

Therefore we will use model 3.31 to characterize the performance of a cluster with static allocation and uniformly probable distribution of ω_0.

Central optimization benchmark

We are asked to solve the constrained minimization problem

$$\begin{cases} T_{req} & = & \sum_{i=1}^n T_i \\ \min_{T_i \in [0, T_{max}]} P_{tot} & = & \sum_{i=1}^n P_{el}(T_i, \omega_i) \end{cases} \tag{3.33}$$

We apply the standard procedure to the simplest case $n = 2$. We can write $T_1 = T - T_2$ hence the minimization is accomplished with the variable T_2

$$\min_{T_2 \in [0, T_max]} P_{tot}(T_2) = P_{el}(T - T_2, \omega_1) + P_{el}(T_2, \omega_2) \qquad T_2 \text{ such that } \frac{\partial P_{tot}}{\partial T_2} = 0 \quad \frac{\partial^2 P_{tot}}{\partial^2 T_2} > 0$$

And we can determine both allocation as

$$T_{1,2} = \frac{T}{2} \pm \frac{\omega_2 - \omega_1}{4} \cdot \frac{k_t}{R} \cdot \left(2B \frac{R}{k_t} + k_v \right) \tag{3.34}$$

Note that:

- If the initial condition are also symmetrical ($\omega_1 = \omega_2$), we return to the static division proving that indeed, in some cases it is the most efficient solution.

- If $\omega_1 \gg \omega_2$, it means that RW_2 would be much more efficient than RW_1. Therefore, $T_2 \gg T_1$ and most of the torque demand is supplied by RW2.

- While performing the optimization we have assumed that all the constant for the two RW are the same ($k_t^{(1)} = k_t^{(2)}, k_v^{(1)} = k_v^{(2)}$ etc) for simplicity. If we do not use this assumption we obtain the allocation 3.35.

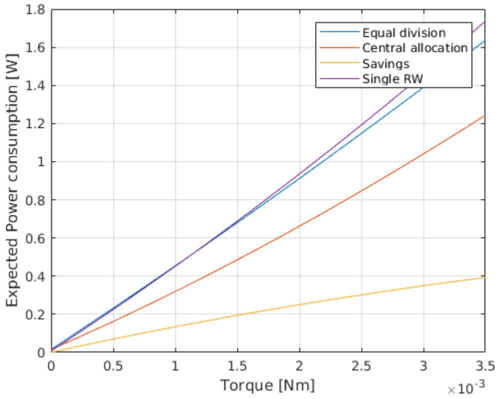
If we allow the two RW to be different, we have

$$T_2 = \left(\frac{R_1}{k_{t,1}^2} + \frac{R_2}{k_{t,2}^2} \right) \left[\frac{2}{k_{t,1}} \cdot T + \frac{w_1}{k_{t,1}} \cdot \left(\frac{2B_1 R_1}{k_{t,1}} + k_{v,1} \right) - \frac{\omega_2}{k_{t,2}} \left(\frac{2B_2 R_2}{k_{t,2}} + k_{v,2} \right) \right] \tag{3.35}$$

The expected power consumption is then

$$\bar{P}_{el}(T_{req})|_n = \int_0^{\omega_{m1}} \int_0^{\omega_{m2}} \left[P_{el}(T_1(T_{req}), \omega_1) + P_{el}(T_2(T_{req}), \omega_2) \right] \cdot fq(\omega_1, \omega_2) \, \mathrm{d}\omega_1 \mathrm{d}\omega_2$$

As the allocation function and the probability density functions become more complex, numerical approaches become the most practical method. In Fig. 3.25-3.26 we compare expected power consumption with static division and central allocation, for a cluster with $n = 2$, under uniform pdf.



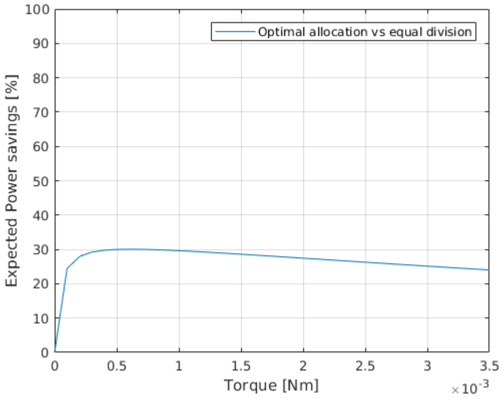
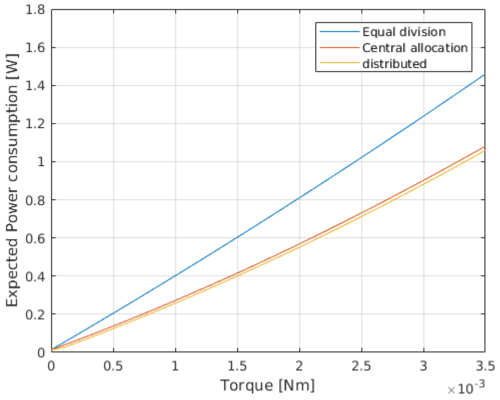


Figure 3.25: Different $\mathcal{P}$ using static division or a central allocator.

Figure 3.26: Percent savings in expected power consumption

3.4.3 Distributed allocation

We consider the distributed optimization scheme proposed in the previous chapter and compare it with static allocation and centralized optimization. Results for the case $n = 2$ are shown in figures 3.27 - 3.28



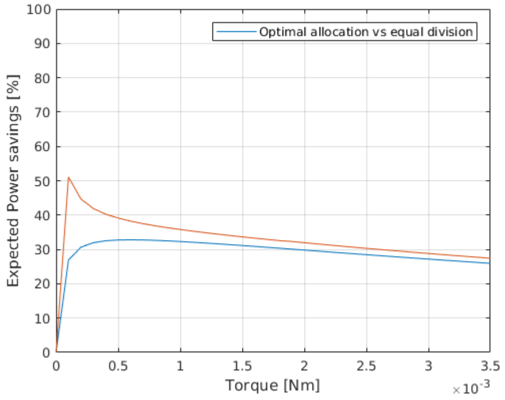


Figure 3.27: Comparison between optimal allocation vs distributed allocation, estimated power requirement with constant pdf

Figure 3.28: Comparison between optimal allocation vs distributed allocation, power savings in % with constant pdf

From Fig. 3.27, the distributed approach (if correctly tuned) exhibits efficiencies which are very close to those of the central allocator. It is interesting to note that efficiency seems higher using the distributed approach. This is merely a result of the numerical method used to implement the dynamical equations. The problem lays in the numerical implementation, which produces a error (in defect) of around 0.02mN-m. The cluster therefore always produces a bit less torque, therefore consuming a bit less power. This approximately constant error impacts more severely the lower end of torque output, as clearly visible in the first part of the efficiency plot in Fig. 3.28. To avoid these deceiving artifacts in the next section we will consider absolute power saved.

3.4.4 Efficiency trends with larger numbers of agents

We go beyond $n = 2$. Plots x are normalized by maximum output to enable comparison among clusters with different authority. Results are presented as the improvement (power saved) compared to the static allocation. This performance is measured from a number of tests and averaged. Each test consists of a target torque level and a vector of randomly selected initial conditions $\vec{\omega}_0$ for the various actuators in the cluster. Using the same target level and initial conditions both static allocation and the proposed algorithm are evaluated. Dual Ascent is not implemented, as it would find the same optimal points as the proposed algorithm[4]. The efficiency attributed to that n and T level is the average of 50 tests.

In these tests, the dynamic of ω is not simulated, because we are interested only in a single torque request/level.

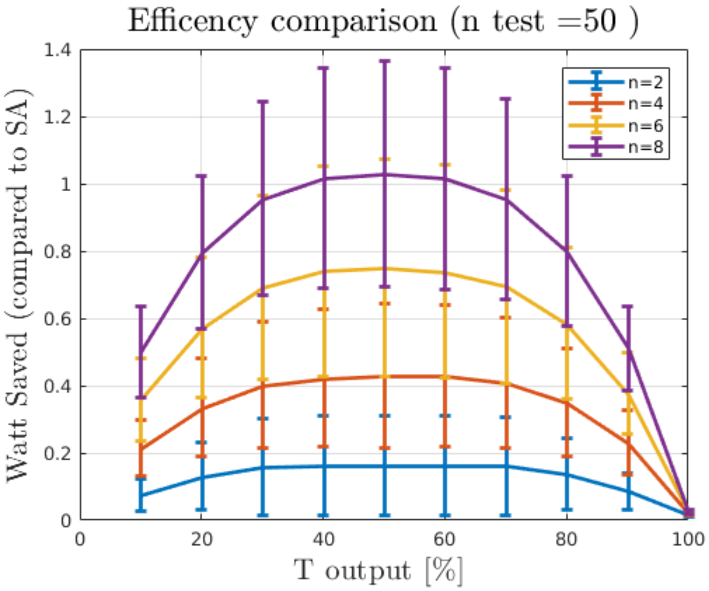


Figure 3.29: Results with the AA model. Parameter for the test in table 3.9

We can note that all curves will end(start) at the same point as there is only one way to produce 100% (0% respectively) of the output.

Using model A0, which has slightly different coefficients, we obtain a different result, as shown in Fig. 3.30. Although the improvements are less pronounced, the overall shape remain similar.

[4] At least for the cost functions with are convex with torque.

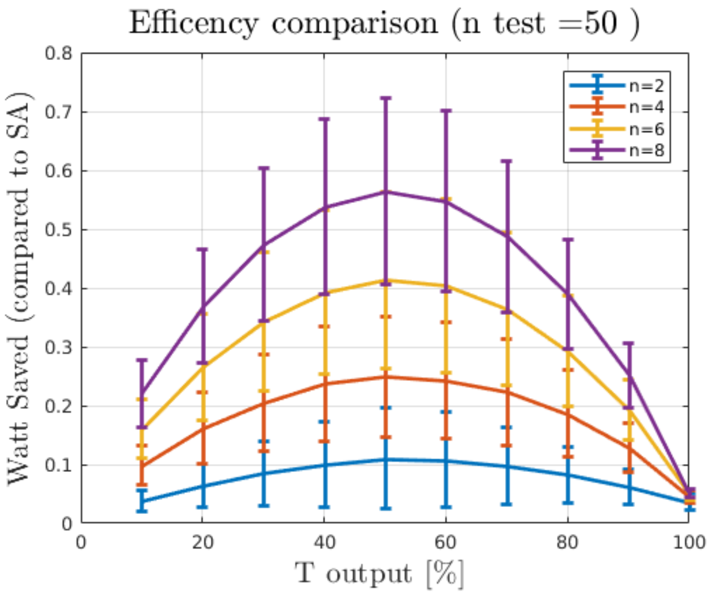


Figure 3.30: Results with the A0 model. Parameter for the test in table 3.9

Finally, the modified model does not deviate significantly from the previous two.

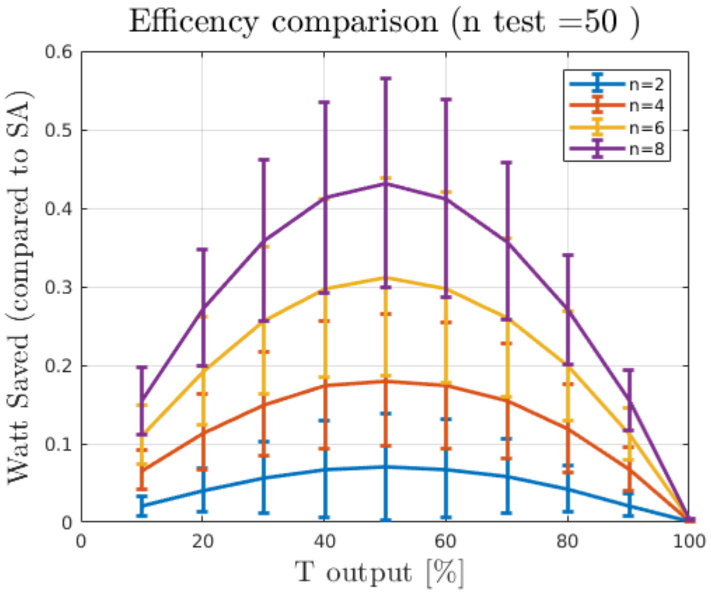


Figure 3.31: Results with the Modified model. Parameter for the test in table 3.9

Model	$t_{\max}$	k	ε	nTest
AA	500	$k_0 \cdot 0.7 = 0.0123$	20	50
A0	500	$k_0 \cdot 0.15 = 0.0156$	100	50
M0	500	$k_0 \cdot 0.9 = 0.0088$	92	50

Table 3.9: Coefficient for the various efficiency study

3.4.5 Robustness under discrete time implementation

Dual Ascent can also be implemented without communication, but without guarantees of convergence when the cost function is not convex. In this section we want to compare Dual Ascent and the proposed method to assess :

- Robustness and convergence speed under discrete time implementation
- Scalability for both methods, namely the effects that increasing the number of agents has on cluster behavior
- Robustness to noise and other non ideal conditions
- Validation of the proposed method to track a dynamic input

We quickly review some details regarding the experiments:

- Impact of ω
  As we discussed at length in the previous sections, the angular velocity of the rotor greatly affects the performance of the RW. To control for this dependency, randomly selected initial conditions will be used and the average results presented. Furthermore $\omega_i(t)$ evolves according to the individual allocation of T chosen by each RW; this evolution is accounted for and can be observed for example in Fig. 3.33 as a slow decrease in efficiency after target output has been reached.
- Test batches
  To statistically control for initial conditions and requested torque, every data point is the results of the average properties of a batch of tests. Every batch is composed of 100 tests, obtained by randomly assigning 10 torque requests and testing each starting from 10 random starting conditions.
- Convergence condition
  We consider that an algorithm has converged at time t_i if the total output of the cluster is within 5% of the requested value and stays within this bound for all the remainder of the simulation. We use the first t_i which verifies this condition as an indication of the speed of convergence. If the maximum number of iteration is reached before this condition is achieved, the algorithm is said to have failed to converge.

Test under ideal condition

Using model A0 (based on the analytical model from rpm/s to W) Fig. 3.33 shows the results of tests with batch from 1 to 20 agents.

Parameters for the proposed methods (k, ε) and Dual ascent (λ_0, α) have been tuned to obtain similar performances when $n = 1$. This is a sensible choice as we can expect that an agent might have to function on its own before joining the cluster; therefore it would make sense to want the individual agent to behave with similar performances regardless of the algorithm used. Furthermore, this criteria fixes the degrees of freedom of the method and thus provides a more fair comparison. The top plot of Fig. 3.32 shows that both methods converge to a good approximation in about 50 iterations regardless of the initial conditions. The bottom plot shows that they reach similar efficiencies after a transitory period. The values for the parameters are reported in 3.10.

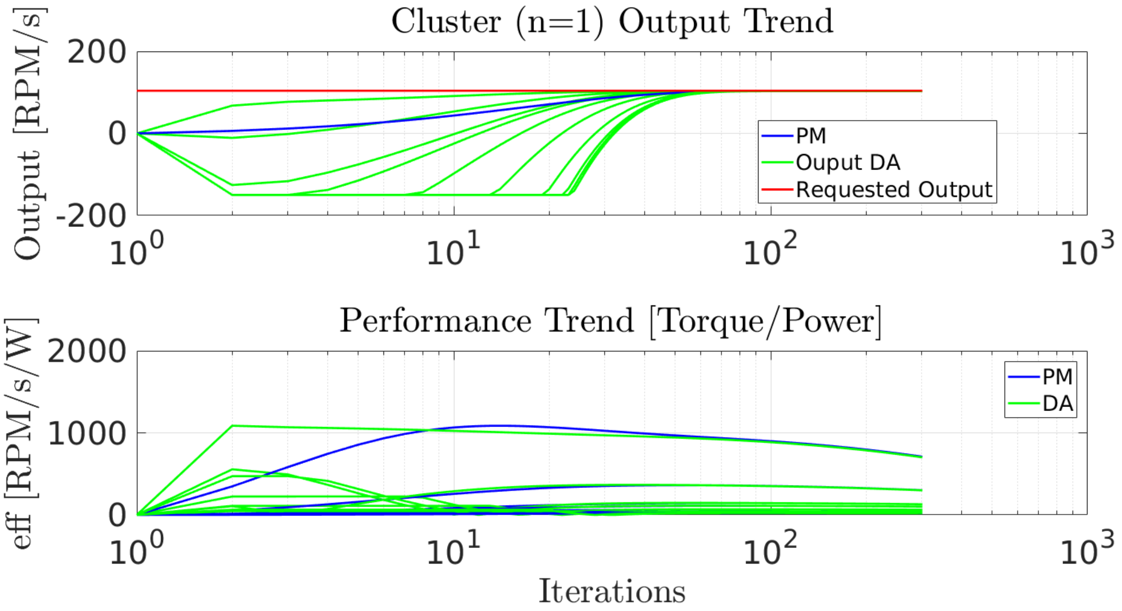


Figure 3.32: A set of tests with a single agent, multiple initial conditions. On the top, convergence to the requested torque, in red. On the bottom, the efficiency plot.

Coefficient	Model	Value
k	Proposed Alg	$75 \cdot k_0 = 0.5849$
ε	Proposed Alg	0.1
α	Dual Ascent	0.0000008
λ_0	Dual Ascent	0.001
t_{max}	both	200
Δt	both	10 ms

Table 3.10: Values used in the simulation with the A0 model for power (and cost).

Now we turn to consider the results portrayed in Fig. 3.33. As a consequence of our tuning strategy, with a small clusters $(n < 5)$ the two methods converge in a similar number of iterations. However DA features significantly higher deviation in convergence time, making it less predictable. This feature is also clearly visible in the top plot of figure 3.32 where, while the DA profiles are visibly separate, the proposed method multiple runs are stacked on top of each other. Between 5 and 10 agents, standard deviation for DA seems to reach a minima. It is interesting to notice that both methods converge more quickly as more agents are added. For the proposed method, this behavior has been predicted by the results on its speed characterization, which we recall here

$$\Delta R(t) \approx \Delta R_0 e^{-nk_0 t} \quad \text{for } \Delta R \gg 1 \tag{3.36}$$

As n increases, we expect convergence to the constraint to speed-up as well. For Dual Ascent however, this feature quickly becomes destructive, preventing convergence for large clusters. This is due to the fact that the discrete time step is fixed; as more agents as added the cluster dynamic speeds up but the numerical implementation remains unchanged. Eventually the time step becomes too big compared with the speed of the cluster and convergence is no longer assured. On the right side of Fig. 3.33, we can see that the deviation for DA convergence starts to increase significantly for $n > 10$ and eventually, convergence is no longer reached. The red dots mark failure to converge in at least one of the 100 tests. This behavior is delayed in the proposed method, which makes it better suited for the control of large clusters.

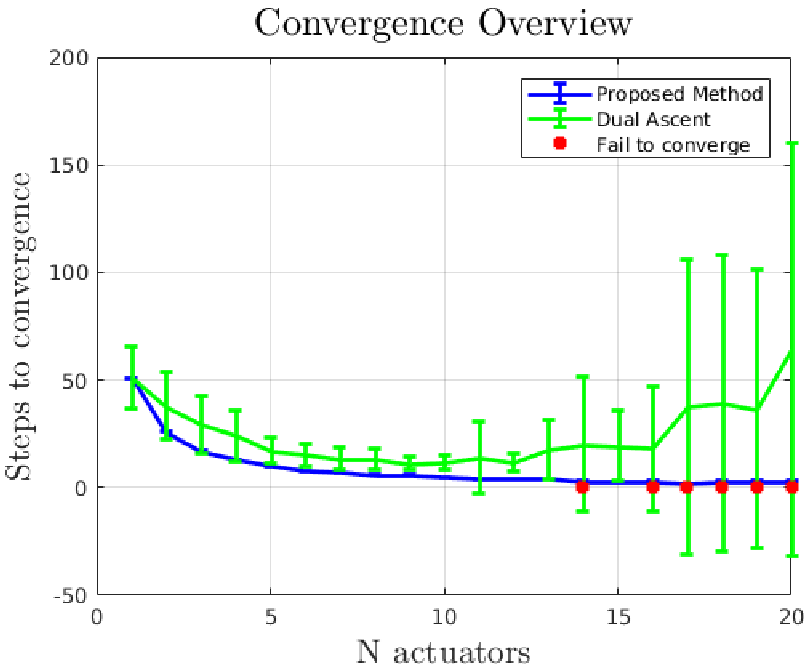


Figure 3.33: Average number of steps to convergence and standard deviation (at 1σ) for the proposed method (in blue) and Dual Ascent (in green). The red dots mark DA batches in which at least one test has failed to converge in the allotted time.

3.4.6 Effect of different λ_0

Dual Ascent depends on a synchronization variable, the Lagrange multiplier λ_0, which in the general implementation is redistributed among the agents. However, since its evolution is entirely governed by the global variable ΔR and the step α of the algorithm, according to

$$\lambda^{i+1} = \lambda_i + \alpha \cdot \frac{\partial \mathcal{L}}{\partial \lambda} = \lambda_i + \alpha \cdot \Delta R \tag{3.37}$$

as long as all agents agree on the same α and start with the same value for λ_0, the evolution of the individual λ will be identical, and thus their value at each time step.

However, as agent might join and leave the cluster in unpredictable ways, different histories might destroy the synchronization. We want to assess this possibility by initializing the various agents from different values of λ_0. The distribution used to randomly select the λ is a uniform pdf within [0,0.002] such that its mean value is the same as in the previous test. Results are shown in figure 3.34. All other parameters for the simulation are the same, as reported in table 3.10.

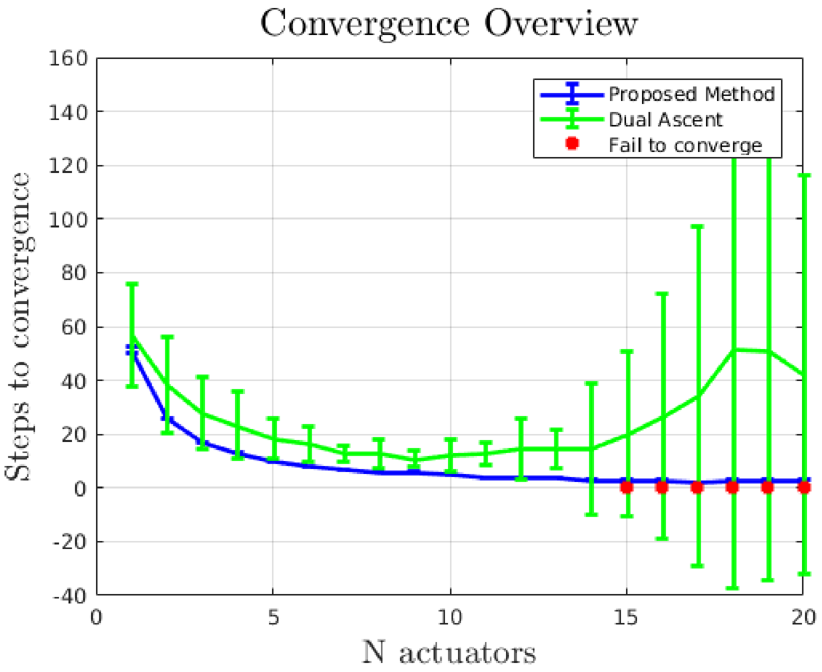


Figure 3.34: Each agent in Dual Ascent is started from different value of λ_0.The proposed method is identical to the ideal case and reported for reference.

Compared with the ideal case, Dual Ascent seems to be slightly slower and with significantly more deviation in convergence speed, but the overall trend does not change. Minor differences in the threshold after which the algorithm ceases to converge can be attributed to the random nature of the numerical experiment.

3.4.7 Effect of random noise

We now perturb the ideal conditions of the initial test with the addition of aggressive levels of noise. Three types of random disturbances are added to the measurement of ΔR; a proportional component, an additive component and a boolean filter. The first is Gaussian, while the other two are based on a uniform distribution. Their contribution is computed as

$$\Delta R_{\text{measured}} = n_3\Big((n_1+1)\cdot\Delta R_{\text{real}} + n_2\Big) \qquad n_3(x) = \begin{cases} x & \text{with probability } 90\% \\ 0 & \text{with probability } 10\% \end{cases} \tag{3.38}$$

The characteristics for the three noise variable n_i are reported in Table 3.11. In this simulation we start all DA agents with the same value for λ_0.

	Noise	pdf	μ	σ
n_1	Proportional	Gaussian	0	10 %
n_2	Additive	Uniform	0	± 5 [RPM/s]
n_3	Packet loss	Uniform		10 %

Table 3.11: Coefficient for the noise functions

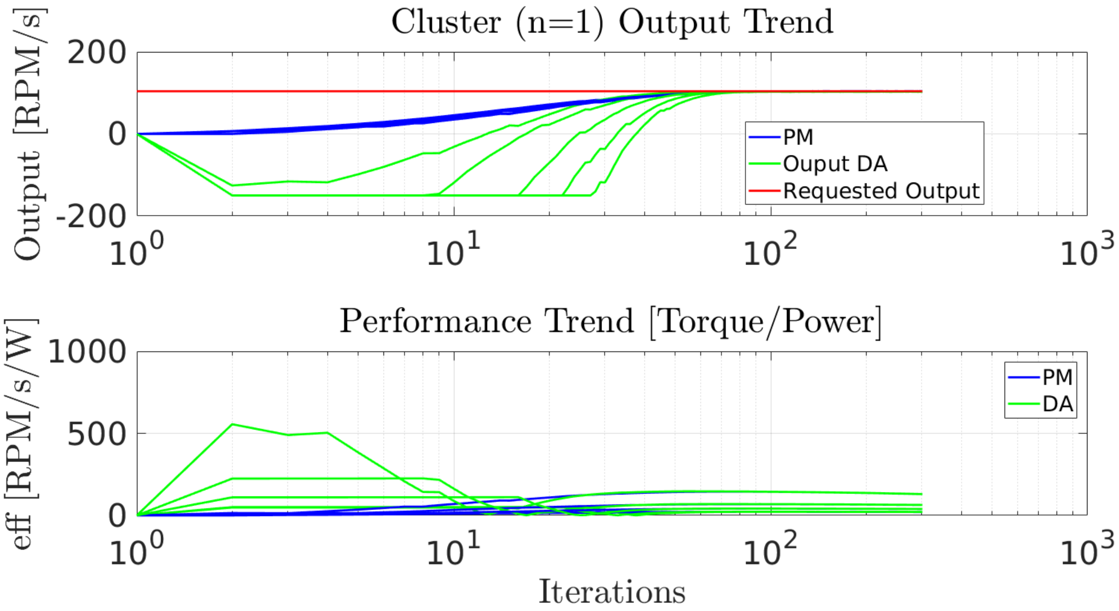


Figure 3.35: Single component, noise as per table3.11, 5 different runs. Profiles are more jagged due to the noise contribution.

The results for the single actuator presented in Fig. 3.35 are in agreement with expectations. Both methods show more erratic behaviors but are not significantly impaired by noisy measurements. Surprisingly, this trends continues even for larger clusters, as shown in Fig3.36. It is interesting to note that failure to converge was no longer observed for fewer than 20 agents. This unexpected effect of noise can be attributed to its attenuating effect on convergence speed, which delays the onset of instabilities.

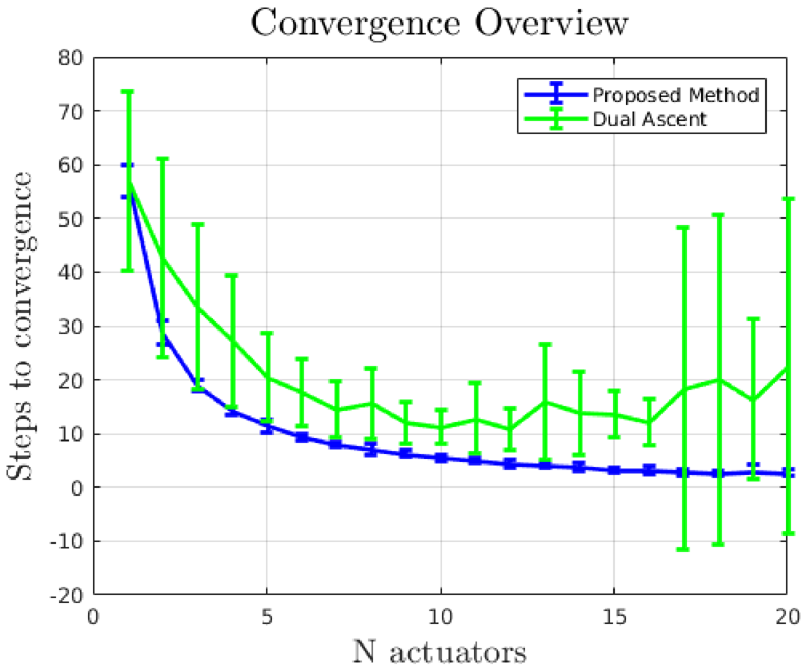


Figure 3.36: 20 Agents, with measurement noise, identical λ_0.

measurement

3.4.8 Dynamic request

Finally we want to merely validate the capability of the proposed method to follow a dynamic signal. Intuitively, assuming that the signal to track changes slowly compared to the cluster dynamic, we can think of it as a static one and apply all the results reported so far. On the other hand, there are many variables to consider. In agreement with the classical characterization of dynamical systems we might wish to study the error in amplitude and timing, or phase shift; we would then track how these parameters vary at different input frequencies and amplitudes (since the model is not obviously linear) thus obtain some numerical approximation of the transfer function.
Considering that, as we have seen in all previous tests, the cluster response changes considerably with the number of agents and the choice of the tuning variables, this procedure would need to be repeated for multiple cases. Finally, the results of this rather extensive set of tests might also reveal a strongly non linear behavior, which would undermine the use of frequency based analysis.

The most sensible way to approach the problem is therefore from the analytical point of view, with proofs and theorems. However, the author was not able to obtain any results and thus this remains an open point. Hence, to at least confirm that the naive approach is a viable operational solution, we present a validation with a cluster of 3 RW tasked to track a sine request. In Fig 3.37, a sine request with a period of 100 timestep is successfully tracked by the cluster. We can observe that both the phase shift and amplitude attenuation seems rather constant.

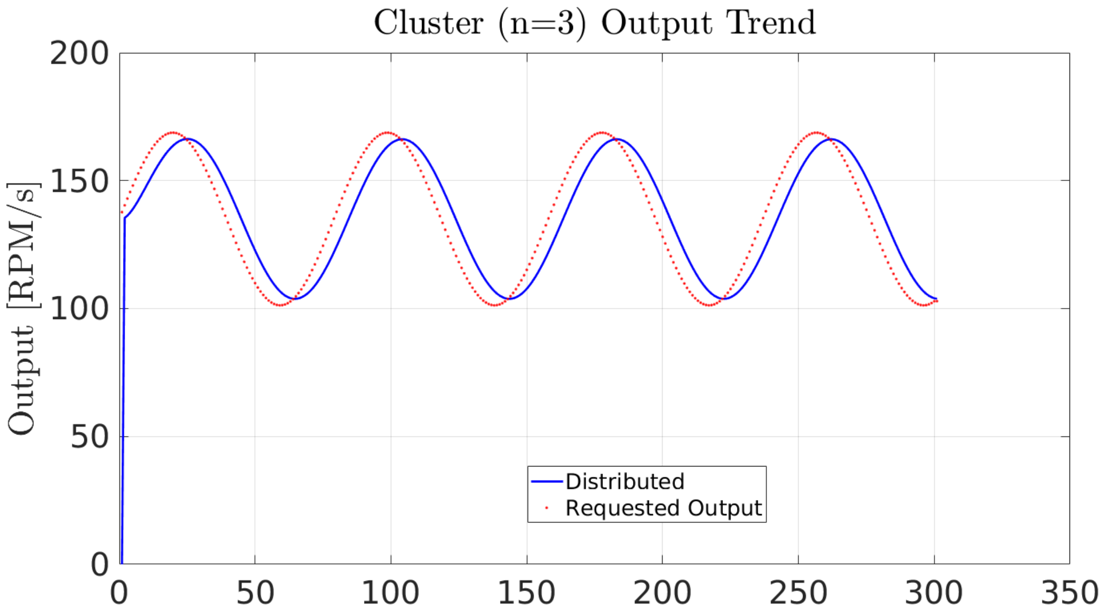


Figure 3.37: A cluster of 3 RW following a target sinusoidal signal at approximately 10 Hz. The units of the x axis are timesteps, each 10 ms long.

Mathematical Details and Calculations

Derivation using the analytical model

Using $P(T,\omega) = aT^2 + Tb\omega + c\omega^2$ as per 3.19 we can solve the the first integral, which models the short term effect. Overall the model is

$$\int_0^{\alpha} P\left(T(t),\omega(t)\right)\mathrm{d}t = \left(\int_0^{\tau} P\left(T_0, \omega_0 + \frac{T_0}{I_w}\cdot t\right)\mathrm{d}t + \int_{\tau}^{\alpha} P\left(T=0, \omega_0 + \frac{T_0}{I_w}\cdot \tau\right)\mathrm{d}t\right)\cdot\frac{1}{\alpha} \tag{3.39}$$

The first integral, dealing with the short term effects:

$$\begin{aligned}
\int_0^{\tau} P\left(T_0, \omega_0 + \frac{T_0}{I_w}\cdot t\right)\mathrm{d}t &= \int_0^{\tau} aT_0^2 + Tb\left(\omega_0 + \frac{T_0}{I_w}\cdot t\right) + c\left(\omega_0 + \frac{T_0}{I_w}\cdot t\right)^2 \mathrm{d}t \\
&= \left| aT_0^2\cdot t + T_0 b\left(\omega_0\cdot t + \frac{T_0}{I_w}\frac{t^2}{2}\right) + c\left(\omega_0^2\cdot t + 2\omega_0\frac{T_0}{I_w}\frac{t^2}{2} + \left(\frac{T_0}{I_w}\right)^2\frac{t^3}{3}\right)\right|_0^{\tau} \\
&= aT_0^2\tau + T_0 b\left(\omega_0\tau + \frac{T_0}{I_w}\frac{\tau^2}{2}\right) + c\left(\omega_0^2\tau + 2\omega_0\frac{T_0}{I_w}\frac{\tau^2}{2} + \left(\frac{T_0}{I_w}\right)^2\frac{\tau^3}{3}\right) \\
&= T_0^2\tau\cdot\left(a + \frac{b\tau}{2I_w} + \frac{c\tau^2}{3I_w}\right) + T_0\tau\cdot\left(b\omega_0 + \frac{c\omega_0\tau}{I_w}\right) + c\omega_0^2\tau
\end{aligned} \tag{3.40}$$

For the long term effects we have:

$$\begin{aligned}
\int_{\tau}^{\alpha} P\left(T=0, \omega_0 + \frac{T_0}{I_w}\cdot\tau\right)\mathrm{d}t &= P\left(T=0, \omega_0 + \frac{T_0}{I_w}\cdot\tau\right)\cdot(\alpha-\tau) \\
&= c\cdot\left(\omega_0 + \frac{T_0}{I_w}\tau\right)^2\cdot(\alpha-\tau) \\
&= \left[\left(\frac{T_0}{I_w}\tau\right)^2 c + \left(\frac{T_0}{I_w}\tau\right)2c\omega_0 + c\omega_0^2\right](\alpha-\tau)
\end{aligned} \tag{3.41}$$

The derivative (needed for both Dual Ascent and the proposed algorithm) becomes:

$$\begin{aligned}
\frac{\partial}{\partial T_0}\int_0^{\alpha} P\left(T(t),\omega(t)\right)\mathrm{d}t = 2T_0\cdot\frac{\tau}{\alpha}\left(a + \frac{b\tau}{2I_w} + \frac{c\tau^2}{3I_w}\right) + \frac{\tau}{\alpha}\cdot\left(b\omega_0 + \frac{c\omega_0\tau}{I_w}\right) + \\
+ \left[2\left(\frac{T_0}{I_w}\tau\right)c + \left(\frac{1}{I_w}\tau\right)2c\omega_0\right]\frac{\alpha-\tau}{\alpha} \\
= 2T_0\cdot\frac{\tau}{\alpha}\left(a + \frac{b\tau}{2I_w} + \frac{c\tau^2}{3I_w}\right) + \frac{\tau}{\alpha}\cdot\left(b\omega_0 + \frac{c\omega_0\tau}{I_w}\right) + \\
+ T_0\frac{2}{I_w}\tau c\frac{\alpha-\tau}{\alpha} + \frac{2c\omega_0}{I_w}\tau\frac{\alpha-\tau}{\alpha}
\end{aligned} \tag{3.42}$$

Derivation using the empirical model

Using $P(T,\omega) = aT + b\omega^2 + c\omega + d$ as per 3.20 we can solve the the first integral, which models the short term effects:

$$\begin{aligned}\int_0^\tau P(T_0,\omega_0+\frac{T_0}{I_w}\cdot t)\mathrm{d}t &= \int_0^\tau aT_0+b\left(\omega_0^2+2\omega_0\frac{T_0}{I_w}\cdot t+\frac{T_0}{I_w}^2\cdot t^2\right)+c\left(\omega_0+\frac{T_0}{I_w}\cdot t\right)+d\;\mathrm{d}t\\ &= aT_0\cdot t+b\left(\omega_0^2\cdot t+2\omega_0\frac{T_0}{I_w}\cdot\frac{t^2}{2}+\frac{T_0^2}{I_w^2}\cdot\frac{t^3}{3}\right)+c\left(\omega_0+\frac{T_0}{I_w}\cdot\frac{t^2}{2}\right)+d\cdot t\Big|_0^\tau\\ &= aT_0\tau+b\left(\omega_0^2\tau+2\omega_0\frac{T_0}{I_w}\frac{\tau^2}{2}+\frac{T_0^2}{I_w^2}\frac{\tau^3}{3}\right)+c\left(\omega_0+\frac{T_0}{I_w}\frac{\tau^2}{2}\right)+d\tau\\ &= \left(\frac{T_0}{I_w}\tau\right)^2\frac{\tau}{3}b+T_0\tau\cdot\left(a+\frac{\omega_0 b}{I_w}\tau+\frac{c\tau}{2I_w}\right)+\tau\left(b\omega_0^2+c\omega_0+d\right)\end{aligned}\tag{3.43}$$

For the long term effect we have:

$$\begin{aligned}\int_\tau^\alpha P\left(T=0,\omega_0+\frac{T_0}{I_w}\cdot\tau\right)\mathrm{d}t &= P\left(T=0,\omega_0+\frac{T_0}{I_w}\cdot\tau\right)\cdot(\alpha-\tau)\\ &= \left(b\left(\omega_0+\frac{T_0}{I_w}\cdot\tau\right)^2+c\left(\omega_0+\frac{T_0}{I_w}\cdot\tau\right)+d\right)\cdot(\alpha-\tau)\\ &= \left[\left(\frac{T_0}{I_w}\tau\right)^2 b+\left(\frac{T_0}{I_w}\tau\right)\cdot(2b\omega_0+c)+(b\omega_0^2+c\omega_0+d)\right]\cdot(\alpha-\tau)\end{aligned}\tag{3.44}$$

Equi probability pdf

Assuming equiprobability for all values of ω, we obtain

$$\bar{P}_{el}(T_{out})=\frac{1}{\omega_{max}}\cdot\int_0^{\omega_{max}}P_{el}(T_{out},\omega)\mathrm{d}\omega$$

Thus integrating

$$\bar{P}_{el}(T_{out})=\frac{1}{k_t\cdot\omega_{max}}\left(\frac{R}{k_t}\cdot T^2w+\frac{1}{2}\left(2B\frac{R}{k_t}+k_v\right)Tw^2+w^3\frac{B}{3}\cdot\left(\frac{BR}{k_t}+k_v\right)\right)|_{0,\omega_{max}}$$

And finally, knowing saturation ω_{max}

$$\bar{P}_{el}(T_{out})=\frac{R}{k_t^2}\cdot T^2+\left(B\frac{R}{k_t^2}+\frac{k_v}{2k_t}\right)w_{max}\cdot T+w_{max}^2\frac{B}{3}\left(\frac{BR}{k_t^2}+\frac{k_v}{k_t}\right)$$

Using a non constant pdf

Let us assume instead that the pdf changes linearly from a starting value p_0 at $\omega=0$ to a final value p_1 at $\omega=\omega_m$

$$fq(\omega)\doteq\frac{1}{\omega_m}\frac{2}{p_0+p_1}\left(p_0+\frac{(p_1-p_0)}{\omega_m}\omega\right)\qquad\Rightarrow\qquad\int_0^{\omega_m}fq(\omega)\,\mathrm{d}\omega=1\tag{3.45}$$

Solving the integral in Eq. 3.28 we obtain

$$P_{el}(T) = \frac{1}{\omega_m}\frac{2}{p_0+p_1}\int_0^{\omega_m} p_0 \cdot P_{el}(T) + \frac{p_1-p_0}{\omega_m}P_{el}(T)\cdot\omega\,\mathrm{d}\omega$$

To ease the computation, we split it into two parts. The first is very similar to the integral obtained in the constant pdf case

$$\frac{1}{\omega_m}\frac{2}{p_0+p_1}\int_0^{\omega_m} p_0\cdot P_{el}(T)\,\mathrm{d}\omega = p_0\cdot\bar{P}_{el}(T) =$$

$$= \left(\frac{2p_0}{p_0+p_1}\right)\cdot\left[\frac{R}{k_t^2}\cdot T^2 + \left(2B\frac{R}{k_t^2}+\frac{k_v}{k_t}\right)\frac{w_{max}}{2}\cdot T + w_{max}^2\frac{B}{3}\left(\frac{BR}{k_t^2}+\frac{k_v}{k_t}\right)\right]$$

The second instead is

$$\left(\frac{2}{\omega_m^2}\frac{p_1-p_0}{p_1+p_0}\right)\cdot\int_0^{\omega_m}\left(T^2\frac{R}{k_t^2}\right)\cdot\omega + \left[T\left(2B\frac{R}{k_t^2}+\frac{k_v}{k_t}\right)\right]\omega^2 + \left(\frac{B^2R}{k_t^2}+\frac{Bk_v}{k_t}\right)\cdot\omega^3\,\mathrm{d}\omega =$$

$$= c_1\cdot\left\{\left(T^2\frac{R}{k_t^2}\right)\cdot\frac{\omega^2}{2} + \left[T\left(2B\frac{R}{k_t^2}+\frac{k_v}{k_t}\right)\right]\frac{\omega^3}{3} + \left(\frac{B^2R}{k_t^2}+\frac{k_v}{k_t}\right)\cdot\frac{\omega^4}{4}\right\}|_{\omega_m} =$$

$$= \left(\frac{2}{\cancel{\omega_m^2}}\frac{p_1-p_0}{p_1+p_0}\right)\cdot\left\{\left(T^2\frac{R}{k_t^2}\right)\cdot\frac{\cancel{\omega_m^2}}{2} + \left[T\left(2B\frac{R}{k_t^2}+\frac{k_v}{k_t}\right)\right]\frac{\omega_m^{\cancel{3}1}}{3} + \left(\frac{B^2R}{k_t^2}+\frac{k_v}{k_t}\right)\cdot\frac{\omega_m^{\cancel{4}2}}{4}\right\}$$

We put them together and obtain ($\Delta p \doteq p_1 - p_0$)

$$\bar{P}(T) = \left(\frac{2}{p_1+p_0}\right)\cdot\left\{\left[\left(\frac{\Delta p}{2}+p_0\right)\frac{R}{k_t^2}\right]\cdot T^2 + \left[\left(\frac{\Delta p}{3}+\frac{p_0}{2}\right)\left(2B\frac{R}{k_t^2}+\frac{k_v}{k_t}\right)\omega_m\right]\cdot T + \left(\frac{\Delta p}{4}+\frac{p_0}{3}\right)\left(\frac{B^2R}{k_t^2}+\frac{k_v}{k_t}\right)\omega_m^2\right\}$$

Analytical Allocation

We apply the standard procedure to the simplest case $n = 2$. We can write $T_1 = T - T_2$ hence the minimization is accomplished with the variable T_2

$$\min_{T_2} P_{tot}(T_2) = P_{el}(T-T_2,\omega_1) + P_{el}(T_2,\omega_2) \qquad T_2 \text{ such that } \frac{\partial P_{tot}}{\partial T_2} = 0 \quad \frac{\partial^2 P_{tot}}{\partial^2 T_2} > 0$$

For instantaneous power consumption, (hence ω doesn't change significantly) we have

$$\frac{\partial P_{tot}}{\partial T_2} = -1\cdot\frac{\partial P_{el,1}(T-T_2,\omega_1)}{\partial T} + \frac{\partial P_{el,2}(T_2,\omega_2)}{\partial T}$$

From Eq. 3.6, we can compute the derivative as

$$\frac{\partial P_{el}}{\partial T} = \frac{1}{k_t}\cdot\left(2T\frac{R}{k_t} + \omega\cdot(2B\frac{R}{k_t}+k_v)\right)$$

Hence, we have

$$-\frac{1}{k_t^{(1)}}\cdot\left(2(T-T_2)\frac{R^{(1)}}{k_t^{(1)}} + \omega_1\cdot(2B^{(1)}\frac{R^{(1)}}{k_t^{(1)}}+k_v^{(1)})\right) + \frac{1}{k_t^{(2)}}\cdot\left(2(T_2)\frac{R^{(2)}}{k_t^{(2)}} + \omega_2\cdot(2B^{(2)}\frac{R^{(2)}}{k_t^{(2)}}+k_v^{(2)})\right) = 0$$

If the two RW are identical $\cdot^{(1)} = \cdot^{(2)}$

$$2(2 \cdot T_2 - T)\frac{R}{k_t} + (\omega_2 - \omega_1) \cdot (2B\frac{R}{k_t} + k_v) = 0$$

And we can determine both allocation as

$$T_2 = \frac{T}{2} - \frac{\omega_2 - \omega_1}{4} \cdot \frac{k_t}{R} \cdot (2B\frac{R}{k_t} + k_v) \tag{3.46}$$

$$T_1 = \frac{T}{2} + \frac{\omega_2 - \omega_1}{4} \cdot \frac{k_t}{R} \cdot (2B\frac{R}{k_t} + k_v) \tag{3.47}$$

Finally, we check the sign of the second derivative

$$\frac{\partial^2 P_{tot}}{\partial^2 T_2} = \frac{\partial^2 P_{el}(T - T_2, \omega_2)}{\partial T_2^2} + \frac{\partial^2 P_{el}(T_2, \omega_2)}{\partial T_2^2} = 2\left(\frac{R^{(1)}}{k_t^{(1)}} + \frac{R^{(2)}}{k_t^{(2)}}\right) > 0$$

Chapter 4

Generalized Clusters

In the previous chapters, we have first explored the advantages of having multiple actuators devoted to the same task and then considered various methods to coordinate their actions. Finally, we have proposed a new decentralized optimization algorithm and we have compared its performances with the methods available in literature. Throughout the exposition, we focused on an architecture based on the repetition of multiple homogeneous actuators. This assumption is especially suited where requirements on reliability explicitly dictates a n-fold redundancy or where due to effects like economy of scale, multiple instances of the same component cost less than a larger, perhaps custom made option.

In this chapter we will weaken this assumption by expanding the idea of redundancy; from the capability to use similar actuators to produce the same output, to the opportunistic collaboration of different systems in the production of a collective set of outputs. What we propose is to use secondary effects as temporary redundancies and to enforce collaboration between heterogeneous actuators, where byproducts from one can decrease the load of another.

The idea is motivated by the observation that real system often produce by-products which could be usefully exploited. While components are typically designed to produce a single output (torque, current, thrust etc), due to inefficiencies they will also generate heat, vibration, EM fields etc. In some cases, careful design can reduce these unwanted contributions or mitigate their impact on the system. In other instances, they may be inescapable traits of the system (such as the heat generated by a CPU). To account for these byproducts, we will need to expand the model for the basic actuator, which so far has been assumed a single input-single output. Thus, we need to allow for multiple inputs, as the type of resources that can be consumed by the actuator, and multiple output, to enable the consideration of byproducts.

4.1 Extending to MIMO

4.1.1 Multiple input generalization

Having proved that the proposed control scheme is generally well behaved, we will refer to it throughout this chapter while approaching the MIMO framework. However, the scheme we use to consider multiple input is a well known one and can be applied to Dual Ascent as well as any other decentralized optimization.

In the SISO framework, we have n parallel components that, using the same resource, produce an homogeneous output, while minimizing the consumption of the resource in question. We have called the cumulative consumption function

$$G(x_1, x_2, \dots, x_n) = g_1(x_1) + \cdots + g_n(x_n)$$

As long as $G() \in C^2$, we can minimize it in a decentralized way. Therefore, if components i, j consume different resources we can reuse the same scheme by establishing an equivalence between the two different resources.

Expanding the consumption vector for each actuator to $\mathbb{R}^m$, allotting one entry of the vector for each possible consumed input, and providing a cost vector $\vec{c} \in \mathbb{R}^{m++}$, we can write

$$G(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n} \vec{c}^{\intercal} \cdot \vec{g}_i(x_i)$$

All the proofs provided so far are still valid, but the actuators no longer need to be homogeneous with regard to the resources they consume.

The problem of the choice of $\vec{c}$ is both interesting and complicated, and will be the core concern of section 4.2.

4.1.2 Multiple outputs for each agents

Previously, agents were assumed to be completely interchangeable, all contributing to the production of the same type of output. We wish to prove similar results of convergence and efficiency for agents which produce an assortment of different goods/outputs.

If the production space is completely disjoint, the extension is trivial, but informative. Let $x_i, i = 1, 2, ...k$ be a set of agent which produces the good R_1, while $x_i, i = k+1, ...n$ produce the good R_2. Then we can write

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_k \\ \dot{x}_{k+1} \\ \vdots \\ \dot{x}_n \end{pmatrix} = k_0 \underset{\mathbf{P}}{\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & \vdots \end{bmatrix}} \left(\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} - \underset{\mathbf{Q}}{\begin{bmatrix} 1 & 1 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & \dots & 1 \end{bmatrix}} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix} \right) - \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix} \tag{4.1}$$

Equation 4.1 is just a repetition; there is no correlation between the production of R_1 and R_2 beside the fact that both consume the same resource. Notice that the matrix $\mathbf{Q}$ plays the role of the vector $\vec{1}^{\intercal}$ and is determined by the system architecture; the j-th column of $\mathbf{Q}$ contains the information about the output for the j-th agent. The information about ΔR is redistributed to the respective agents through the matrix $\mathbf{P}$.

Like in the mono dimensional case, if $\Delta R = \vec{R} - \mathbf{Q}X = 0$, the system is producing what it is supposed to, and therefore it is approximately in equilibrium (if we neglect the contributions of the g_i).

A more interesting case can be made when the two productions are not disjoint. In general we can imagine a cluster of n agents, which all use the same resource, and can collectively produce $m < n$ independent amounts of goods. We will show that the decentralized approach is still able to converge to the constrained optima.

Consider the following control law

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \vdots \\ \dot{x}_n \end{pmatrix} = k_0 \begin{bmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,m} \\ p_{2,1} & p_{2,1} & \dots & p_{2,m} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ p_{n,1} & p_{n,m} & \dots & p_{n,m} \end{bmatrix} \left(\begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \end{pmatrix} - \begin{bmatrix} q_{1,1} & q_{1,2} & \dots & \dots & q_{1,n} \\ q_{1,1} & q_{2,2} & \dots & \dots & q_{2,n} \\ \vdots & & & & \vdots \\ q_{m,1} & q_{m,1} & \dots & \dots & q_{m,n} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} \right) - \begin{pmatrix} \frac{\partial G}{\partial x_1} \\ \frac{\partial G}{\partial x_2} \\ \vdots \\ \vdots \\ \frac{\partial G}{\partial x_n} \end{pmatrix} \tag{4.2}$$

Or, in a more compact form

$$\dot{X} = k_0 \mathbf{P} \left(\vec{R} - \mathbf{Q} \cdot X \right) - \nabla G \qquad \begin{matrix} \mathbf{P} \in \mathbb{R}^{n \times m} \\ \mathbf{Q} \in \mathbb{R}^{m \times n} \end{matrix} \tag{4.3}$$

$\mathbf{Q}$ is a general matrix imposed by the physics of the system; we assume it to be full row rank[1]. If this was not the case, we could not produce any arbitrary output[2]. Then, $\Delta R = R - \mathbf{Q}X$ still retains the idea of *a set of* collective measurement. Matrix $\mathbf{P}$ is a degree of freedom, and is chosen among the generalized inverse of $\mathbf{Q}$; in particular, it needs to be a right inverse, meaning to satisfy

$$\mathbf{P} = \mathbf{Q}_R^{-1} \qquad \text{such that} \qquad \mathbf{Q} \cdot \mathbf{P} = \mathbb{I}_m \tag{4.4}$$

The vector $\mathbf{P} \cdot \Delta R$ informs each agent of the action needed to reach $\mathbf{Q}X = \vec{R}$. For a quick intuition of why $\mathbf{P}$ moves the system in the right direction, towards the satisfaction of the constraint, observe that for $k_0 = 1$ and ∇G negligible

$$\begin{cases} X_0 & = & \vec{0} \\ \Delta X & \approx & \mathbf{P} \cdot \vec{R} \end{cases} \Rightarrow \begin{cases} X_1 & \approx & \cancel{X_0} + \Delta X \\ \Delta R_1 & = & \vec{R} - \mathbf{Q}X_1 & = & \vec{R} - \cancel{\mathbf{QP}} \cdot \vec{R} = \vec{0} \end{cases} \tag{4.5}$$

A canonical way to find a right inverse for $\mathbf{Q} \in \mathbb{R}^{m \times n}$ is [3]

$$\mathbf{P} = \mathbf{Q}^\intercal \left(\mathbf{Q}\mathbf{Q}^\intercal \right)^{-1} \qquad \Rightarrow \qquad \mathbf{Q} \cdot \mathbf{P} = \mathbf{Q} \cdot \mathbf{Q}^\intercal \left(\mathbf{Q}\mathbf{Q}^\intercal \right)^{-1} = \mathbb{I}_m \tag{4.6}$$

In the proofs we will use a more specific $\mathbf{P}$, which can always be derived from any right inverse. Let $p_1, p_2, \dots p_m$ be the columns of the canonical $\mathbf{P}$; by definition 4.4 we have that individually

$$\mathbf{Q} \cdot p_i = \vec{e}_i \qquad \hat{e}_i = (0, \dots, \underset{i}{1}, \dots, \underset{m}{0})^\intercal \tag{4.7}$$

We require that, for each vector v in the kernel of $\mathbf{Q}$, we have $v \cdot p_i = 0$. This is easily achieved though a Gram-Schmidt based argument. Let $v_1, \dots v_j, p_1, \dots, p_k$ be a basis for the domain of $\mathbf{Q}$,

[1] A matrix is full row rank when the rows of the matrix are linearly independent.

[2] If two rows are linearly dependent, their product with any $\vec{x}$ will be linearly dependent, so we loose the freedom to produce any $R \in \mathbf{R}^m$

[3] Existence of $\mathbf{P}$ is guaranteed by the assumption that $\mathbf{Q}$ has full row rank

where $v_i \in \ker(\mathbf{Q})$. By the Gram-Schmidt process we obtain an orthonormal base which respects the same division $\hat{v}_1, \ldots \hat{v}_j, \hat{p}_1, \ldots, \hat{p}_k$. Using the $\hat{p}_i$ vector, we can obtain a temporary matrix $\mathbf{P}'$ which we can use to combine the $\hat{p}_i$ into the wanted vector. For each $\hat{e}_i$ we find a suitable $\vec{\alpha}$ such that

$$\mathbf{Q} \cdot \left[\begin{pmatrix} \hat{p}_1 \end{pmatrix}, \ldots, \begin{pmatrix} \hat{p}_k \end{pmatrix} \right] \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \ldots \\ \alpha_k \end{pmatrix} = \hat{e}_i \tag{4.8}$$

The various $\mathbf{P}'\vec{\alpha}_i$ will form the columns of the $\mathbf{P}$ we wanted; orthogonality with the kernel is preserved by linearity in 4.8.

An algebraic alternative

An algebraic method to obtain the matrix $\mathbf{P}$ is also possible. We note that adding any vector v belonging in the kernel of $\mathbf{Q}$ to any vector p_i does not alter condition 4.7. Hence we can always adjust the p_i to bring some projection to zero. Calling v_j with $j = 1, \ldots, J$ the vectors in the kernel of $\mathbf{Q}$, we define

$$p_i' = p_i + \sum_{k=1}^{J} \alpha_k \cdot v_k \tag{4.9}$$

Imposing that the product with each v is zero we have a system of J equations in J unknowns.

$$\begin{array}{rcl} v_1 \cdot p_i + \alpha_1 \cdot v_1 v_1 + \alpha_2 \cdot v_1 v_2 + \cdots + \alpha_J \cdot v_1 v_J & = & 0 \\ v_2 \cdot p_i + \alpha_1 \cdot v_2 v_1 + \alpha_2 \cdot v_2 v_2 + \cdots + \alpha_J \cdot v_2 v_J & = & 0 \\ \ldots & & \\ v_J \cdot p_i + \alpha_1 \cdot v_J v_1 + \alpha_J \cdot v_1 v_2 + \cdots + \alpha_J \cdot v_J v_J & = & 0 \end{array} \tag{4.10}$$

Solving for the coefficients α we obtain a suitable p_i; repeating for all p_i we obtain a matrix $\mathbf{P}$ with the desired properties.

LaSalle, weak proof

Everything is similar to the monodimentional case, except when we prove that the equilibrium points where $\dot{X} = \vec{0}$ are stationary points for $G()$.

As a Lyapunov function, we again choose $V(X)$ such that its derivative with respect to time is a simple quadratic function

$$\frac{\partial V}{\partial t} \doteq -\dot{X}^{\intercal} \cdot \dot{X} = -\sum_{i=1}^{n} \dot{x}_i^2 \tag{4.11}$$

Which guarantees that $\dot{V} \leq 0$ for all $X \in \chi$ and that, for every $X \in \chi$ such that $\dot{V} = 0$, all components of $\dot{X}$ must be zero.

$$\frac{\partial V}{\partial t} = 0 \quad \Leftrightarrow \quad \dot{x}_i = 0 \quad \forall i = 1, 2, ..., n \tag{4.12}$$

Again, since $\frac{\partial V}{\partial t} = \nabla V \cdot \dot{X}$ we can satisfy Eq. 4.11 by imposing

$$\begin{pmatrix} \frac{\partial V}{\partial x_1} \\ \frac{\partial V}{\partial x_2} \\ \dots \\ \frac{\partial V}{\partial x_n} \end{pmatrix} = -\dot{X} = -k_0 \mathbf{P} \vec{R} + k_0 \mathbf{P} \mathbf{Q} X + \vec{\nabla} G \tag{4.13}$$

It is easy to check that this differential condition is satisfied by the flowing Eq. 4.14

$$V(X) \doteq -k_0 (\mathbf{P}\vec{R})^\intercal \cdot X + \frac{k_0}{2} \vec{1}^\intercal \cdot \mathbf{PQ} \cdot X^{\times 2} + G(X) \tag{4.14}$$

Where $X^{\times 2} \doteq (x_1^2, \dots, x_n^2)^\intercal$. Which is $C^1(\chi, \mathbb{R})$ since $G \in C^2(\chi, \mathbb{R})$. Then we have satisfied all the hypothesis of the invariance principle, and therefore $X(t)$ will converge to the largest positively invariant set $\mathcal{M}$ contained in $\mathcal{Z} = \left\{ X \in \chi \text{ such that } \dot{V}(X) = 0 \right\}$.

At equilibrium, we have that

$$\dot{V} = 0 \quad \Leftrightarrow \quad k_0 \mathbf{P}(\vec{R} - \mathbf{Q} X) = \nabla G \tag{4.15}$$

In order to prove that a stationary point for the dynamic of the system is also a constrained stationary point for the consumption $G(\cdot)$, we need to prove that the directional derivative of $G(\cdot)$ along any direction on the constrain hyperplane is zero.

$$\forall \vec{s} \text{ in the constraint} \qquad D|_s G = \nabla G \cdot \vec{s} = 0$$

By definition, the perturbation $\vec{s}$ is compatible with the constraint if its application does not change ΔR, the distance from the constraint. Hence

$$\vec{s} \text{ such that } \Delta R(X) = \Delta R(X + \vec{s}) \quad \Leftrightarrow \quad \vec{R} - \mathbf{Q} \cdot X = \vec{R} - \mathbf{Q} \cdot (X + \vec{s}) \quad \Leftrightarrow \quad \mathbf{Q}\vec{s} = \vec{0}$$

The kernel of $\mathbf{Q}$ and the co domain of $\mathbf{Q}$ divide $\mathbb{R}^n$ in two complementary sub spaces. We have chosen $\mathbf{P}$ such that all its columns are vector orthogonal to the vector in kernel of $\mathbf{Q}$. Hence, every vector ΔR through $\mathbf{P}$ will be a linear combination of vectors perpendicular to the kernel and therefore perpendicular to any $\vec{s}$ on the constraint. Then we have that

$$\vec{0} = \left(k_0 \mathbf{P}(\vec{R} - \mathbf{Q} X) \right)^\intercal \cdot \vec{s} = \nabla G^\intercal \cdot \vec{s} = D|_s G \tag{4.16}$$

Then we have proven that any equilibrium for the dynamic of the system is a stationary point for the cost function $G(\cdot)$.

Stability of the minima for G.

Exactly as in the mono dimensional case,

$$V(X) = \left(\frac{\partial}{\partial t} X \right)^2 = f(X)^2 \tag{4.17}$$

Its derivative is then

$$\frac{\partial}{\partial t}V(X) = (2 \cdot f(X) \cdot \nabla f(X)) \cdot \frac{\partial}{\partial t}X = 2f(X) \cdot \nabla f(X) \cdot f(X)$$

However, now $\nabla f(X)_{i,j} = \frac{\partial}{\partial x_j} f(X)_i$ is

$$\nabla f = -k_0 \mathbf{PQ} - [\nabla^2 G] \tag{4.18}$$

We need to prove that an equilibrium point Ω is stable if it is a constrained minima for G. To prove the stability, we need to show that $\dot{V}$ is negative in a closed set around Ω. We prove this by showing that $\mathbf{A} \doteq -[\nabla f(X)]$ is definite positive in a neighborhood of Ω. If $\mathbf{A}$ is definite positive in Ω, by continuity we can find a small ball in which it is still definite positive.

Given that Ω is a constrained minimum, $\nabla^2 G$ is definite positive in the direction of the constrain. Since any vector on the constrain hyper-plane is in the kernel of $\mathbf{Q}$, we have

$$\vec{s}^{\intercal} \cdot (k_0 \mathbf{PQ} + [\nabla^2 G])\vec{s} = k_0 \vec{s}^{\intercal} \cdot \mathbf{P}\cancel{\mathbf{Q}\vec{s}} + \vec{s}^{\intercal} \cdot [\nabla^2 G])\vec{s} > 0 \tag{4.19}$$

On the other hand, if $\vec{s}$ is not on the constraint, it can be written as a sum of p_i vector hence $\vec{s} = \mathbf{P}\vec{\alpha}$

$$\begin{array}{rcccc} \vec{s}^{\intercal} \cdot (k_0 \mathbf{PQ} + [\nabla^2 G])\vec{s}^{\intercal} & = & k_0 \cdot \vec{s}^{\intercal} \mathbf{P}(\mathbf{QP})\vec{\alpha} & + & \vec{s}^{\intercal}[\nabla^2 G]\vec{s} \\ & = & k_0 \cdot \vec{s}^{\intercal} \mathbf{P}\vec{\alpha} & + & \vec{s}^{\intercal}[\nabla^2 G]\vec{s} \\ & = & k_0 \cdot \vec{s}^{\intercal}\vec{s} & + & \vec{s}^{\intercal}[\nabla^2 G]\vec{s} \\ & = & k_0 \cdot \|\vec{s}\| & + & \vec{s}^{\intercal}[\nabla^2 G]\vec{s} \end{array} \tag{4.20}$$

Which is always greater than zero if we choose a sufficiently large k_0, according to

$$k_0 > \min_{i \in 1,\dots,n} \frac{\partial^2 G}{\partial x_i^2} \tag{4.21}$$

Instability of the maxima for G

We will use Chetaev theorem. We need to show that, for some appropriate $V \in \mathcal{C}^1(\chi)$, there is (at least) a continuous path characterized by $V > 0$ and $\dot{V} > 0$ that starts in the stationary point Ω and exits the neighborhood $\mathcal{N}_r(\Omega)$. Our $V(\cdot)$ function is a distance from the equilibrium point X_Ω, which is always positive except in X_Ω

$$V(X) = (X - X_\Omega)^2 \qquad \frac{\partial V}{\partial t} = 2(X - X_\Omega)(f(X) - f(X_\Omega)) \tag{4.22}$$

Since X_Ω is an equilibrium point and considering a small movement $\vec{s} = X - X_\Omega$ we can write (using Taylor approximation)

$$\frac{\partial V}{\partial t} = 2\vec{s} \cdot (\cancel{f(X_\Omega)} + \nabla f(X_\Omega) \cdot \vec{s}) = -2\vec{s}(k_0 \mathbf{PQ} + \nabla^2 G(X_\Omega))\vec{s} \tag{4.23}$$

Considering $\vec{s}$ in the constraint direction $\mathbf{Q}\vec{s} = 0$, we have that

$$\frac{\partial V}{\partial t} = -2\vec{s}\,\mathbf{P}\cancel{\mathbf{Q}\vec{s}}k_0 - 2\vec{s}\,\nabla^2 G(X_\Omega)\vec{s} = -2\vec{s}\,\nabla^2 G(X_\Omega)\vec{s} \tag{4.24}$$

If X_Ω is a maximum along the constrained direction, $\vec{s}^\intercal \nabla^2 G|_{X_\Omega}\vec{s} < 0 \Rightarrow \dot{V} > 0$. Hence, in the neighborhood of a constrained maxima there is a clear escape route along the constraint. Moving in any direction $\vec{s}$ on the constraint will increase the value of V, and retain a positive $\dot{V}$.

If X_Ω is a saddle point, there must be at least one direction in which the product $\vec{s}\nabla^2 G\vec{s}$ is negative, otherwise it would be constrained minima. Therefore the same reasoning applies, but not in all directions on the constraint.

Remarks

We have extended the decentralized optimization scheme to both multiple inputs and multiple outputs model. A few remarks:

- To allow multiple outputs to be considered, each agent needs to know at least its respective row of the matrix **P**, which depends on matrix **Q**. Structural information about the system architecture is thus required.
- We are minimizing an artificial cost function of our own choosing, so it might or might not bear physical meaning.
- All systems are assumed to be controlled by a single variable; by-products are assumed to be linear with the main output.

4.2 An endogenous cost function

There are cases in which a cost function is externally provided and is not arbitrary. For example, a system which uses gas and electricity from the grid can compare them through the economic cost, their *dollar value*. In this section we will try to come up with a reasonable cost function that can be used when external ones are not provided.

The cost function is a convenient tool to achieve some secondary goal; primary goals on the other hand, are set with constraints. Thus we begin by considering which outcome we wish to obtain and then we try to design a cost function accordingly. Typically, an autonomous system such as a satellite or a rover, has some onboard reservoir of resources, such as batteries, fuel tanks, momentum wheels and so on. These reservoirs have finite capacity and every action that the system takes will consume some of their capacity. We propose to assign value to internal resources based on their usefulness for the system. However, once the system is able to produce the outputs we want, we need to establish a criteria to assess usefulness.

We suggest to maximize the life of the system, as determined by the time for which the system is able to respond to our request before running out of some essential resource. We refer to this measure as responsive time.

We defend this choice with the following remarks:

- It is based only on system and mission considerations; we are not adding external preferences or assessments. If we are able to achieve maximization of responsive time, it would be a parsimonious choice from the point of view of assumptions.

- Many missions are to some extent temporally open ended, such as rovers on mars, the ISS or earth observation satellites; the nominal mission is short but can be extended if the hardware is still able to support it. In such cases, responsive time is a natural choice.
- In the case of missions with a clearly defined time horizon, such as a launcher or an interplanetary transfer vehicle, maximizing responsive time is not interesting in itself. However, it can be viewed as the dual of the minimization of initial amount of resources. With simulations one can minimize leftovers at the end of mission, and thus improve the design.
- Any objective function needs to be a real valued function as $\mathbb{R}^n$ does not have nice ordering properties. Hence mapping choices in $\mathbb{R}^n$ onto their consequence on a time duration is a mathematically sound choice.

We begin by formally defining the problem. We call $\phi \in \mathbb{R}^n$ the vector the operational levels of the n agents, collectively referred as the *action* of the system. For the sake of simplicity, we bound it within $[0,1]^n$, where a 0 values on the kth term signifies that the kth system is off while 1 signifies that the system is at maximum output. We call $\mathbf{B}(\phi)$ the function that maps an action to its output (the torque, current, thrust it produces) and $\mathbf{E}(\phi)$ the function that maps an action to the resources needed to accomplish it.

Then we wish to find a solution $\phi(t) : \mathbb{R}^+ \to [0,1]^n$ that meets requirements $\vec{R}(t)$ (Eq. 4.25) and doesn't exceed initial resources $\vec{I}_0$ (Eq. 4.26) for the longest possible time, t_f.

$$\mathbf{B}(\phi(t)) = \vec{R}(t) \tag{4.25}$$

$$\forall t < t_f \qquad \vec{I}_0 - \int_0^t \mathbf{E}(\phi(t))\mathrm{d}t \gg \vec{0} \tag{4.26}$$

It won't go unnoticed that to solve the problem we could very well avoid a cost function altogether. Two conceptually simple strategies which follow this approach are considered briefly and rejected as both are affected by the same drawbacks

- A pragmatic approach would be to come up with a set of functions $\phi_i(t)$ that satisfy $\mathbf{B}(\phi(t)) = \vec{R}(t)$ by design, compute their responsive time and choose the best. The discriminant between the candidates is the choice of some proper inverse $\mathbf{B}^{-1}(\cdot)$, which applied to $\vec{R}(t)$ defines $\phi(t)$ and therefore consumption $\mathbf{E}(\phi(t))$. Then, integrating consumption we can detect when a resource exceeds the initial amount and record the responsive time for each candidate. The best will be the chosen over some discretization of the function space of $\mathbf{B}^{-1}$, as the one that induces the ϕ associated with the longest t_f.

  By testing more candidates we can increase the confidence that the best option we found was a global optimum but the computational burden also increases. In general however, we do not know $\vec{R}(t)$ beforehand and, in line with the decentralized approach typical of clusters, we might wish to avoid centralized computation and coordination.
- To avoid the computational burden of a discretized approach, one could pursue an analytical solution. If such solution exists, it would seem likely for it to depend on the request $R(t)$, but this remains a conjectures as we were unable to make any progress in this direction. However, even if such a one shot, request- independent solution was found, it would still be a centralized method, which we wish to avoid.

To better support decentralization, we choose to investigate an indirect approach; the agents are set up to individually minimize a cost function which has been designed to promote behaviors which maximize responsive time. We loosely govern the system from a high level, relying on the individual agents to perform the detailed optimization.

The relationship between cost and responsive time can be broken down as follows: cost drives the agents choices in consumption, the more something costs the less it will be used if an alternative is available. By shaping the cost function we provide preferential directions in the resource space. Clearly, the position of the system in the resource space is linked to responsive time, as the latter is defined as the instant in which we cross a boundary of the positive resource quadrant. If we were able to develop some *longer term* relationship between resource space and responsive time, we could use it to design of the cost function.

The outcome of this strategy is a *just in time* approach, which does not use knowledge of $\vec{R}(t)$ and can be implemented in a decentralized manner. The drawback of decoupling high level objectives and low level detail is that we abandon the hope of knowing the exact actions of the individual agents; consequently, closed form results seems very hard to obtain. Nevertheless, for special cases of $\mathbf{E}(\cdot)$ and $\mathbf{B}(\cdot)$ we provide a set of observations about responsive time maximization. The main results is that, for linear models, reaching the origin is associated with maximum responsive time. Hence, we can define a cost function which pushes the system towards the origin. Outside the linear case, the problem seems quite more complex and we were unable to find any results beyond mere conjectures.

Linear case

In the simplest SISO model, efficiency is constant. To generalize to MIMO, we impose the ratio between any output and any input to be constant. The system behaves linearly, with constant *generalized* efficiencies. For every pair of input-output types,

$$\frac{\text{output}_i}{\text{input}_j} = \varepsilon_{i,j}(\text{output}) = \text{constant} \qquad \text{output}_i = \varepsilon_{i,j} \cdot \text{input}_j \tag{4.27}$$

Eq 4.27 translates to

$$\text{output vector} = \begin{pmatrix} o_1 \\ o_2 \\ \vdots \\ o_r \end{pmatrix} f(x) \qquad \text{input vector} = \begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_m \end{pmatrix} f(x) \qquad \mathcal{E} = \begin{bmatrix} {}^{o_1}/_{i_1} & {}^{o_1}/_{i_2} & \dots & {}^{o_1}/_{i_m} \\ {}^{o_2}/_{i_1} & \dots & \dots & {}^{o_2}/_{i_m} \\ \vdots & \vdots & \vdots & \vdots \\ {}^{o_r}/_{i_1} & \dots & \dots & {}^{o_r}/_{i_m} \end{bmatrix} \tag{4.28}$$

Notice that $\mathcal{E}$ is defined to satisfy $\mathcal{E} \cdot \vec{i} = \vec{o}$, which fits the intuitive concept of efficiency[4]. Under this model, generalized efficiency does not change with the operational level x of the system.

As usual, the function $f(x)$ could be a generic monotone function [5]. However, we can choose to transform it into linear model without loss of generality. Formally, we are controlling the value

[4] However, the elements can not be taken individually; the input space in this model is $\vec{i} \cdot s$, $s \in [0,1]$, not the whole space $\mathbb{R}^m$

[5] The operational level x can be interpreted as the physical variable that conveys the command to operate the system, such as a reference voltage, which the agent uses to determine how much to produce.$f(x)$ can always be assumed increasing monotone. This is how systems are usually made, but it is easy to show that is always possible to make it so.

of x through some knowledge of the system behavior

$$x = g(y) \quad g \doteq f^{-1}(y) \quad y \in [0,1] \tag{4.29}$$

And we know that a preferred f^{-1} exists. From now on, we will assume that $f(x) = x$ and $x \in [0,1]$, to ease the notation.

4.2.1 Model notation and hypothesis

Consider a request space $\mathbb{R}^r$, and a resource space $\mathbb{R}^m$. The system has n agents, which collaborate to produce a vector of outputs in the request space according to the linear function

$$\vec{R} = \vec{o}_1 \cdot x_1 + \ldots \vec{o}_n \cdot x_n \quad \begin{cases} \mathbf{B} & \doteq & \left[\vec{o}_1, \ldots, \vec{o}_n\right] \\ \\ \Phi & \doteq & (x_1, \ldots, x_n)^\intercal \end{cases} \quad \Rightarrow \quad \vec{R} = \mathbf{B} \cdot \Phi \tag{4.30}$$

In the compact form, $\mathbf{B} \in \mathbb{R}^{r \times n}$ is the matrix that describes what the system is producing. How the agents collaborate to meet requirement is encoded in this matrix. In the previous chapter we called this $\mathbf{Q}$, but since the objective of this section is different, we deemed appropriate to maintain separate names.

The overall consumption, which can be interpreted as a movement in the resource space, is also a linear function of the operational state vector Φ

$$\mathbf{E} \doteq \left[\vec{i}_1, \ldots, \vec{i}_n\right] \quad \Rightarrow \quad \mathbf{E} \cdot \Phi = -\dot{I} \tag{4.31}$$

The matrix $\mathbf{E} \in \mathbb{R}^{m \times n}$ expresses system consumption for a given Φ, the amount of resources which is collectively consumed to produce the output $\mathbf{B} \cdot \Phi$.

We assume that both $\mathbf{B}$,$\mathbf{E}$ and a generalized inverse of $\mathbf{B}$ are populated only by non negative terms. Note that to ensure that the generalized inverse $\mathbf{B}_g$ exists we need to require that $\mathbf{B}$ has full row rank. To justify these assumptions, we observe that:

- Requiring $\mathbf{B}$ to be component wise positive has an immediate engineering connotation, conveying the idea that all agents cooperate. Since

  $$\mathbf{B} \cdot \Phi \gg \vec{0} \qquad \forall \Phi \gg 0 \tag{4.32}$$

  Where the $\gg$ relationship symbolizes strict inequality in a component-wise sense; $\vec{x} \gg \vec{y} \Leftrightarrow x_i > y_i \, \forall i$. We understand that the system is somehow *conservative*, all action will lead to consequences in the request space and there is no way to *waste* any action; no system can undo what another is doing.

- Requiring $\mathbf{E}_{i,j} \geq 0 \, \forall i, j$ means that no agent can generate resources; all allowable actions $\Phi \gg \vec{0}$ will consume some amount of resources.

- Requiring $\mathbf{B}$ to be full row rank guarantees that the system is able to produce any request $\vec{R}$. This is consistent with the idea that any output for the system has at least an agent dedicated primarily to its production.

- We call $\mathbf{B}_g$ a generalized inverse of $\mathbf{B}$ (since $\mathbf{B}$ has full row rank, we know it has a right inverse). One such matrix can be defined as

$$\mathbf{B}_g = \mathbf{B}^\intercal(\mathbf{B}\mathbf{B}^\intercal)^{-1} \quad \Rightarrow \qquad \Phi(t) = \mathbf{B}_g\vec{R}(t) + (\mathbb{I} - \mathbf{B}_g\mathbf{B})w \tag{4.33}$$

  In general, due to the application we are studying, we need to require more than full row rank on $\mathbf{B}$; either we require that a generalized inverse exits such that all its components are non negative or we limit the request $\vec{R}(t)$ to ensure that $\mathbf{B}_g\vec{R}(t)$ is always non negative. If this were not the case, there would be requests which could not be met unless an actuator was operated in reverse, producing instead of consuming and vice versa, which is not allowable for the model.
  For example, if we consider

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad \mathbf{B}_g = \begin{bmatrix} 1 & -1 \\ 0 & 1/2 \\ 0 & 1/2 \end{bmatrix}$$

  We would have no way of producing $R = (0,1)^\intercal$ without reversing the production of the first actuator. We will assume that $\mathbf{B}_g$ has only non negative components[6].

Given a request $\vec{R}(t)$, using a generalized inverse $\mathbf{B}_g$ we can easily characterize the allowable movements in the resource space. We express the resource consumption of all the actions ϕ which satisfy the request by expanding $\mathbf{E} \cdot \phi$,

$$-\dot{I} = \mathbf{E} \cdot \phi = \mathbf{E}(\mathbf{B}_g\vec{R} + (\mathbb{I} - \mathbf{B}_g\mathbf{B})w) \quad w \in \mathbb{R}^n \text{ s.t. } \phi \gg \vec{0} \tag{4.34}$$

Where w needs to be chosen so that $\Phi \gg 0$. Notice that, for every $w \in \mathbb{R}^n$, $(\mathbb{I} - \mathbf{B}_g\mathbf{B})w$ is in the kernel of $\mathbf{B}$. Therefore the movements in the resource space are, at most, defined by the image of output request $\mathbf{B}_g \cdot \vec{R}$ and the kernel of $\mathbf{B}$; we can write

$$-\dot{I} = \mathbf{E}\mathbf{B}_g\vec{R} + \mathbf{E} \cdot k_i \qquad k_i \in \text{kernel}(\mathbf{B}) \tag{4.35}$$

4.2.2 Example with positive monomial E

We begin by considering a simplified case for which clear results are easy to obtain. We will extend the idea to a more generic $\mathbf{E}$ once we have developed a clear strategy.

In an extremely simplified case, each agent consumes a unique and dedicated resource. Then $\mathbf{E}^\star$ is a diagonal matrix with positive components (formally called a positive monomial matrix)

$$\mathbf{E}^\star = \begin{bmatrix} e_1 & 0 & 0 & \dots & 0 \\ 0 & e_2 & 0 & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & \dots & e_n \end{bmatrix} \qquad e_1, e_2, \dots e_n > 0 \tag{4.36}$$

For such matrices we can easily correlate the position in the resource space, which is the amount of resources still available for the system to consume, to the total order in the responsive

[6] Assuming $\mathbf{B}_g \gg 0$ is easier conceptually and in the following proofs, but it ultimately depends on the nature of the actuators we use. For practical purposes instead, it might be easier to accept that not all $\vec{R}$ can be requested; it translates into accepting that some byproducts might be inevitable due system architecture.

time domain. This provides a strong indication of where it is most convenient to be to maximize responsive time.

Lemma 8. Given $\mathbf{E}^\star$ positive monomial, $\Phi_1(t) : [0, t_1] \to \mathbb{R}^{n+}$ and $\Phi_2(t) : [0, t_2] \to \mathbb{R}^{n+}$ such that

$$\mathbf{B}\Phi_1(t) = \vec{R}(t) \quad \forall\, t \in [0, t_1] \qquad \mathbf{B}\Phi_2(t) = \vec{R}(t) \quad \forall\, t \in [0, t_2] \tag{4.37}$$

The following holds

$$\int_0^{t_1} \mathbf{E}^\star \Phi_1(t) \mathrm{d}t \ll \int_0^{t_2} \mathbf{E}^\star \Phi_2(t) \mathrm{d}t \quad \Rightarrow \quad t_1 < t_2 \tag{4.38}$$

The counter intuitive interpretation of this lemma is that, as long as we meet the requirement $\vec{R}(t)$, consuming more resources is associated with a longer responsive time. Hence, to achieve maximum responsive time we should *aim* for the origin of the resource space. If we are able to reach the origin, we would know that we have consumed the maximum possible amount of resources (all which were available), and therefore achieved maximum t_f.

Proof. Since $\mathbf{E}^\star$ is clearly invertible, and $\mathbf{E}^{\star\,-1}$ has also strictly positive components placed exclusively along the diagonal

$$\int_0^{t_1} \mathbf{E}^* \Phi_1(t) \mathrm{d}t \ll \int_0^{t_2} \mathbf{E}^* \Phi_2(t) \mathrm{d}t \quad \Leftrightarrow \quad \int_0^{t_1} \Phi_1(t) \mathrm{d}t \ll \int_0^{t_2} \Phi_2(t) \mathrm{d}t$$

We now multiply both side by $\mathbf{B}$. This preserves the partial order relationship since all components are non negative (both in Φ_i and in $\mathbf{B}$).

$$\Rightarrow \quad \int_0^{t_1} \mathbf{B}\Phi_1(t) \mathrm{d}t \ll \int_0^{t_2} \mathbf{B}\Phi_2(t) \mathrm{d}t \quad \Leftrightarrow \quad \int_0^{t_1} \vec{R}(t) \mathrm{d}t \ll \int_0^{t_2} \vec{R}(t) \mathrm{d}t$$

Which can be true only if $t_2 > t_1$. (When we assume that $\vec{R}(t)$ is component-wise strictly positive $\forall t$, to avoid ambiguities). □

Given the current position in the resource space, we can visualize system consumption over a time interval Δt as the vector connecting current position I_0 with the future one $I_{\Delta t}$. Among all the possible options we could chose as end points, the one that is assured to provide the longest responsive time is the one that maximizes $I_0 - I_{\Delta t}$.

For example, in Fig, 4.1, between point A and B, the best choice is clearly to target point A, as it consumes more resources, hence lasting more. Notice that, since we do not have full control of the system, the two options are not to be understood as *move to A* or *move to B*; the correct interpretation is *set a cost that, when minimized, leads to A* or B.

By the same principle, among all the possible targets in the resource space, the origin holds the promise of maximum responsive time. At first glance, this result might seem counter-intuitive, as it could be read as "the more the system consumes, the more it lasts". In fact, the causality relationship is inverted; if the system functions for more time, it will consume *more* resources. Furthermore, it might be the case that reaching other points in the resource space will yield the same responsive time as the origin; Lemma 8 only states that there can not be longer responsive time (given $\vec{R}(t)$).

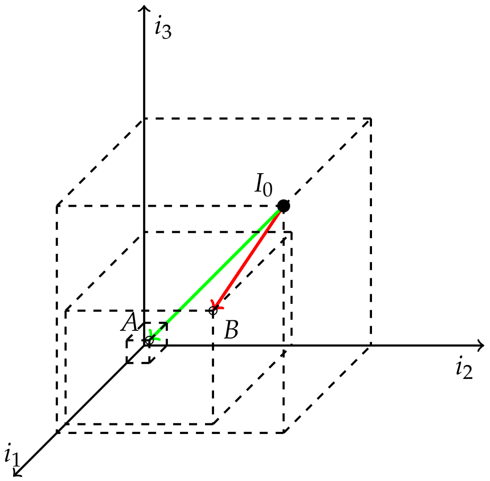


Figure 4.1: Two possible target in the resource space

Is it always possible to reach the origin?

Even in the simplified case considered above, it is not clear when we are able to reach the origin. After all, the primary concern is to produce the correct amount of output $\vec{R}(t)$.

Notice that if $\mathbf{E}$ is not singular, as in the case of strictly positive monomial $\mathbf{E}^*$, linearly independent vectors in the domain will have linearly independent images. Therefore, there are *directions* in the resource space which we can not access with any choice of k (as per Eq. 4.35 , k is a vector in the kernel of $\mathbf{B}$). Movements along such directions are determined by requests $\vec{R}$ and we can not hope to counter act them with any usage of the degrees of freedom of the system.

$$\vec{I}(t) = \vec{I}_0 - \underbrace{\mathbf{EB}_g \int_0^t \vec{R}(\tau)\mathrm{d}\tau}_{\text{imposed motion}} - \underbrace{\mathbf{E}\int_0^t \vec{k}(\tau)\mathrm{d}\tau}_{\text{controllable motion}} \qquad \vec{k}_i \in \ker(\mathbf{B}) \tag{4.39}$$

To highlight this separation, we define an orthogonal basis of the resource space such that the first r elements generate the subspace containing the image through $\mathbf{E}$ of the column of $\mathbf{B}_g$

$$\hat{e}_1,\ldots,\hat{e}_r \in \mathbb{R}^n \text{ such that } \quad \forall\,\vec{x} \in \mathbb{R}^r,\ \mathbf{EB}_g x = \sum_{i=1}^{r} \alpha_i\hat{e}_i \quad \hat{e}_i\cdot\hat{e}_j = 0\,\forall\, i\neq j \tag{4.40}$$

and complete the orthogonal base with $\hat{e}_{r+1}\ldots\hat{e}_n$. Since $\mathbf{E}$ has full rank, $< \hat{e}_{r+1}\ldots\hat{e}_n >$ is a base for the image of the kernel of $\mathbf{B}$. We can define the change of base as

$$\mathbf{S}\doteq \begin{bmatrix} \hat{e}_1{}^{\mathsf{T}} \\ \hat{e}_2{}^{\mathsf{T}} \\ \ldots \\ \hat{e}_n{}^{\mathsf{T}} \end{bmatrix} \qquad S^{-1} = S^{\mathsf{T}} = [\hat{e}_1,\ldots,\hat{e}_r,\hat{e}_{r+1},\ldots,\hat{e}_n] \tag{4.41}$$

and separate the imposed and controllable dynamic

$$\vec{I}(t) = \mathbf{S}^{-1}\cdot\left(\mathbf{S}\vec{I}_0 - \mathbf{S}\mathbf{E}\mathbf{B}_g\int_0^t \vec{R}(\tau)\mathrm{d}\tau + \mathbf{S}\mathbf{E}\int_0^t \vec{k}(\tau)\mathrm{d}\tau\right) \tag{4.42}$$

$$\vec{I}(t) = \mathbf{S}^{-1}\cdot\left(\begin{pmatrix} I'_{0,1} \\ \vdots \\ I'_{0,r} \\ I'_{0,r+1} \\ \vdots \\ I'_{0,n}\end{pmatrix} - \begin{pmatrix} R_1(t) \\ \vdots \\ R_r(t) \\ 0 \\ \vdots \\ 0\end{pmatrix} - \begin{pmatrix} 0 \\ \vdots \\ 0 \\ k_1(t) \\ \vdots \\ k_{n-r}(t)\end{pmatrix}\right) \tag{4.43}$$

This shows that, even if we imagine the simplified scenario, where we neglect the constraint $\Phi(t) \gg \vec{0}$ and we are free to choose the $k_1(t), ...k_{n-r}(t)$, not all end states are reachable. In particular, since **S** is non singular, reaching the origin, $\vec{I}(t) = \vec{0}$ requires particular conditions on I_0 and $\vec{R}(t)$, which are outside our control.

4.2.3 Case for general E

In general, each type of actuator might use more than one resource; we can not expect **E** to be monomial. In many cases, the number of agents easily exceeds the number of resources, so even assuming that **E** has unique inverse would be restrictive. In such cases it is not immediate that reaching the origin would yield the longest responsive time. In general, different vectors in the domain of **E** might be projected onto the same vector in the co-domain. Hence, the decomposition into *controllable* and *uncontrollable* part according to 4.39 is also not as straight forward.

The generalization of Lemma 8 is

- If $\vec{R}(t)$ admits a strategy $\vec{k}(t)$ that, from initial point I_0, reaches the origin, this will happen only once at a unique time $t^\star$. We prove this results with some additional engineering assumptions about the way in which the system is set up.
- If $t^\star$ exists and is unique, it is the longest possible responsive time.

We provide proof following the order of the claims;

Proof. We want to find if there can be multiple t such that $I(t) = \vec{0}$. The proof will fork in three cases depending on the relationship between **B** and **E**. We proceed by reductio ad absurdum. Assume there exists t_1, t_2 where $t_1 < t_2$ such that both can reach the origin through the strategies $\vec{K}_1, \vec{K}_2 \in \ker(\mathbf{B})$. This can be stated as

$$\begin{cases} I(t_1) &= \vec{0} &= I_0 - \mathbf{E}\mathbf{B}_g \cdot \int_0^{t_1} \vec{R}(\tau)\mathrm{d}\tau - \mathbf{E}\vec{K}_1 \\ I(t_2) &= \vec{0} &= I_0 - \mathbf{E}\mathbf{B}_g \cdot \int_0^{t_2} \vec{R}(\tau)\mathrm{d}\tau - \mathbf{E}\vec{K}_2 \end{cases} \tag{4.44}$$

We examine the difference between the two

$$I(t_1) - I(t_2) = \mathbf{E}\mathbf{B}_g \cdot \int_{t_1}^{t_2} \vec{R}(\tau)\mathrm{d}\tau - \mathbf{E}\left(\vec{K}_1 - \vec{K}_2\right) = \vec{0} \quad\Rightarrow\quad \mathbf{E}\mathbf{B}_g \int_{t_1}^{t_2} \vec{R}(\tau)\mathrm{d}\tau = \mathbf{E}\Delta\vec{k} \tag{4.45}$$

We want to prove that the last statement of 4.45 is either impossible or both side are zero, thus proving that $t_1 = t_2$ against the initial assumption.

Case for E non singular The resources consumed in the time elapsed between t_1 and t_2 are entirely compatible to some movement $\Delta\vec{k}$, which being a linear combination of vectors of the kernel of **B**, is also contained in the kernel. On the other hand $\mathbf{B}_g\vec{R}(t)$ lives in the subspace of $\mathbb{R}^n$ generated by the r columns of $\mathbf{B}_g$. Vectors in this space are linearly independent of the kernel of **B**. Then, if **E** is not singular, condition 4.45 is clearly not possible, as two sub spaces which are linearly independent in the domain space of a non singular **E** remain linearly independent in its image space.

Case for $\mathbf{ker}(\mathbf{B}) \subset \mathbf{ker}(\mathbf{E})$ If the kernel of **B** is contained in the kernel of **E**, the right side of 4.45 is zero and therefore the integral has value 0. This can happen only if $t_1 = t_2$, due to hypothesis about **E** ($\gg 0$) and about $\vec{R}(t) \geq 0\ \forall t$.

This may seem as a rather convenient and artificial case from the mathematical point of view, however it carries engineering significance. Consider a system made by selecting different types of actuators (RW, heater, Radio, OBDH etc) to support all the functions needed by the mission. Due to byproducts, there is some overlap in the overall production however the set of production vectors of the selected type is linearly independent. Then, either to increase reliability, throughput, or both, some of the components are used in multiple copies. They are nominally the same component, with the same technical specification but are used in multiples. Since the output vectors of the original type were linearly independent, the only vectors in the kernel of **B** are those in which pairs of identical components are run one against the other. However, since they are identical, their consumption vector will also be the same. Then all vectors in the kernel of **B** will also belong to the kernel of **E**.

This argument is based on the consideration that, in engineering, it is easier to use multiple copies of the same component rather than having to interface with different objects which do the same thing.

Otherwise We decompose the request between its projection onto the kernel space of **B** and the remainder, which we call $\vec{c}$, in the pre image of **B**

$$\mathbf{B}_g \int_{t_1}^{t_2} \vec{R}(\tau)\mathrm{d}\tau = \vec{k}_0 + \vec{c} \qquad \vec{k}_0 \in \ker(\mathbf{B}), \quad \Rightarrow \quad \begin{cases} \vec{c} & \notin \quad \ker(\mathbf{B}) \\ \vec{c} & \in \quad \ker(\mathbf{E}) \end{cases} \tag{4.46}$$

if $\vec{k}_0$ is the projection on $\ker(\mathbf{B})$, $\vec{c}$ is determined unequivocally and must be fully contained in the sub space of the kernel **E** to satisfy the condition 4.45. We consider $\vec{k}_0$, which is not in the kernel of **E** by hypothesis. Its image through **E** must be strictly positive, as

$$\vec{k}_0 + \vec{c} \gg 0 \quad \Rightarrow \quad \mathbf{E}\cdot(\vec{k}_0 + \vec{c}) \gg 0 \quad \text{but } \vec{c} \in \ker\mathbf{E} \quad \Rightarrow \quad \mathbf{E}\cdot\vec{k}_0 \gg \vec{0} \tag{4.47}$$

But this implies that the system design has a serious engineering flaw, as if we divide k_0 into its positive and negative part such that $k_0 = k^+ - k^-$, with $k_i^+, k_i^- \geq 0\ \forall i$ we get

$$\mathbf{B}k_0^+ = \mathbf{B}k_0^- \qquad \text{but} \qquad \mathbf{E}k_0^+ \gg \mathbf{E}k_0^- \gg \vec{0} \tag{4.48}$$

Therefore, we are allowing the system the possibility of producing the exact same amount of output in two ways, one of which consumes strictly more than the other. This should not be a choice, and it is assumed to be prevented by design. This is sensible for most systems, but is especially crucial if we are designing a decentralized one, in which we have relinquished direct control over the actuators.

□

Assuming t^* exists and is unique, we want to prove that if the system has reached the origin, it has achieved the longest possible response time.

We use a special case to expose the core of the general argument. Assume that the system structure is set up so to guarantee complete and absolute interchangeability in the resources used; any request $\vec{R}$ could be met with any single type of resource.

$$\forall \vec{R} \in \mathbb{R}^{r+}, j = 1, \dots m \quad \exists \phi \gg \vec{0} \text{ such that } \mathbf{B}\phi = \vec{R} \text{ and } \mathbf{E}\phi = \alpha \cdot \hat{e}_j \quad \alpha \in \mathbb{R}^+ \tag{4.49}$$

In such case, it does not matter how we instantaneously decide to use the resources. If we reach a boundary of the resource space, running out of a particular resource, we can still respond to any request $\vec{R}$ by using any of the remaining ones. This strategy systematically consumes one resource after the other and the *run* ends when we reach the origin.

Notice that all runs can eventually reach the origin, and since you can only reach the origin at time $t^\star$, $t^\star$ must be the longest possible responsive time. In fact, to achieve any different t_{stop}, you would have to stop before reaching the origin hence $t_{\text{stop}} < t^\star$.

In more realistic cases, several restrictions will prevent us from always reaching the origin. However, if we are able to reach the origin, we might recognize how this is the only *run* which can not be improved, and hence the best.

Proof. Assuming $t^\star$ exists and is unique, we move according to some strategy $\vec{k}(t)$ and reach a boundary of the resource space. There are only two possible cases;

1. We are able to respond to the next request $\vec{R}(t^+)$ without using the depleted resource

2. We can not; we have discovered the responsive time for the strategy $\vec{k}(t)$

In the first case, we proceed until we reach another boundary, at which point we face the same options. Eventually we will fall into the second case, and no longer be able to respond to requests. Let's imagine that we are not in the origin; we no longer have sufficient resources to meet requirements, but we still have positive leftovers of some unused one. We might regret our choices so far, and backtrack. If we had known the request function beforehand, we could have used less of the depleted resources and more of the resources which we have left. Assume we do so, and this is the only change we do on the run. Then we will have improved the responsive time. At this point we can repeat the backtracking with the new information about which resource is more useful for this particular request.

Since we assumed that the origin could be reached, this process will reach it. At this point, we can not longer apply the scheme to increase responsive time. Since the response time of any other point in the resource space can be improved except for the origin, this is the optimum point.

We are forced to conclude that if we can reach the origin, this is the place that yields maximum responsive time. □

Therefore, if we can reach the origin, it is often our best bet, assuming that the system is properly designed. We note that these results are a posteriori; we do not know a prior if the pair $\vec{R}(t)$, I_0 allows us to reach the origin. However, if we fail to reach the origin, we can still claim that we did better than all the other options which ended with strictly more leftover resources.
Consider two runs, A, B, both starting at I_0 and ending at I_A, I_B respectively. Then if $I_A \gg I_B$, it must be that $t_A < t_B$. If we virtually subtract I_B from the initial condition, both path remain valid, but B reaches the origin. Therefore t_B is the longest responsive time of the two. If we had subtracted I_A on the other hand, the run B would not have been legal.

4.2.4 Design of the cost function

At least in the linear case, we should have sufficient motivation to believe that striving to reach the origin is a sensible choice to maximize responsive time. We set out to design a cost function that steer the system towards it.

We begin by observing how the system changes its consumption in response to cost. The cost function is chosen as the scalar product of the cost vector and resource consumption

$$\mathcal{C}(\phi) = \vec{c}^{\mathsf{T}} \cdot \mathbf{E}\phi \tag{4.50}$$

where the j-th component of $\vec{c}$ represent the value we assign to the j-th resource. We can safely assume that all resources have some value, so $c_i > 0$ for all components; otherwise we would encourage waste. Although the exact action ϕ is intrinsically linked to $\vec{R}(t)$ as the optimization is constrained to the anti-image of the request, here we are only interested in the general trend, so we neglect the specific value of $\vec{R}$. To meet request, the system will consume resources and thus cost will always be positive. Then, depending on the effectiveness of the optimization strategy, cost will be positive but small, as close to zero as possible. Geometrically this means that the system will try to move perpendicularly to the cost vector. In Fig. 4.2 we can use this geometrical idea to understand that A costs less than B and, if both options are feasible, we can expect the system to move towards A.

While there may be many different ways to reach the origin from the middle of the resource space, the set of available choices shrinks as the system moves towards a boundary of the positive hyperoctant. To avoid consuming a resource that is already scarce, we would like to move parallel to the boundary as we approach it. Hence we expect the cost vector to be normal to the boundaries of the positive hyperoctant.

The economic-flavored analogue is to set the value of a resource to be inversely related to its availability; the less of a resource remains, the more valuable it will be and the less it will be used. Then we impose the cost of each resource as inversely proportional to the amount available;

$$\vec{c} = \left(\frac{1}{I_1}, \frac{1}{I_2}, \dots, \frac{1}{I_n} \right)^{\mathsf{T}} \tag{4.51}$$

In accord with the decentralized framework, each agent only needs to be aware of the value of each resource or to be able to directly measure the remaining quantity directly and there is no need for centralized coordination. Moreover, resource availability can be easily measured and it is naturally suited to handle resupply dynamics. When new resources are introduced in the system, the cost function changes accordingly.

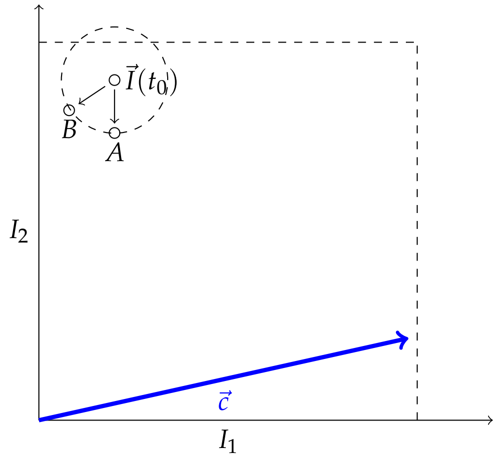


Figure 4.2: Assuming both option A and B feasible, A costs a lot less than option B since it has a smaller projection on the cost vector $\vec{c}$. Note that the arrow represent the movement in the resource space, which is $-\mathbf{E}\phi$; the sign is negative because resources are being consumed.

Simplified model Building upon the geometric intuition, we now show that, for a simplified model, the chosen cost function leads the system to the origin. As previously mentioned, the outcome of the cost minimization is dependent on the request $\vec{R}(t)$, hence we will not obtain strong general results. After all, we know from the previous section that not all combinations of $(\vec{R}(t), \vec{I}_0)$ admits a path to the origin. An even if one was available, it could be very dependent on the knowledge of $\vec{R}(t)$, condition which we do not admit.

To take out the influence of $\vec{R}(t)$, we follow the dynamic of a system which effectively only pursues cost minimization. This assumption means that the toy-system can always satisfy request by choosing among a simplex of the canonical base, provided that it consumes resources. Then, the available choices span the whole negative space;

$$\vec{p} = -(\alpha_1 \cdot \hat{e}_1 + \cdots + \alpha_n \cdot \hat{e}_n) \qquad \text{where } \sum_{i=1}^{n} \alpha_i = 1 \quad \alpha_i > 0 \; \forall i \tag{4.52}$$

Given the cost function 4.51 which depends on the current position of the system $I(t) = (I_1, I_2, \ldots, I_n)^\intercal$, the direction that minimize cost is the one associated with the highest amount of resources

$$\mathcal{C}(\vec{p}) = \frac{\alpha_1}{I_1} + \frac{\alpha_2}{I_2} + \cdots + \frac{\alpha_n}{I_n} \quad \Rightarrow \quad \mathcal{C}(\vec{p}) \text{ is minimized by } \begin{cases} \alpha_i = 1, & \alpha_j = 0 \; \forall j \neq i \\ \text{where} & I_i \geq I_j \; \forall j = 1, \ldots n \end{cases} \tag{4.53}$$

Figure 4.3 shows a 2D implementation of the above strategy. The cost vectors are shown in blue while the optimal movement is in green. Along the diagonal, the system is free to chose any direction, while near the boundaries, cost becomes perpendicular to the chosen, suggesting minimization of the objective function.

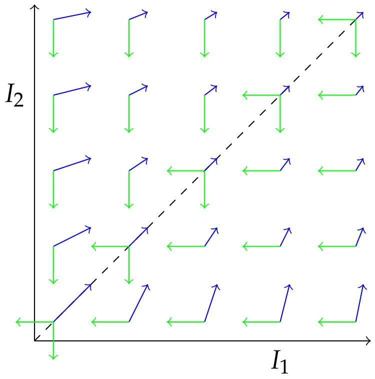


Figure 4.3: Vector field of cost function (in blue) and cost minimization strategy (in green).

The toy-model provides an idea of what happens when the request allows for optimization. In this interpretation, $\vec{R}$ can be viewed as a disturbance to the trajectory which would otherwise end in the origin.

Even if we are not able to reach the origin, minimizing cost 4.51 is sometime equivalent to minimizing resources consumption. To clarify this statement, we provide the following weak result.

Weak result

Consider a $\phi(t)^\star$ among the feasible paths ϕ which minimizes cost $\mathcal{C}(\cdot)$ for all t

$$\mathcal{C}(\phi(t)^\star) \leq \mathcal{C}(\phi(t)) \quad \forall \phi \text{ such that } \mathbf{B} \cdot \phi(t) = \vec{R}(t) \tag{4.54}$$

Calling t_f the response time of $\phi^\star$, we have that

$$\forall t_1 < t_f \quad \int_0^{t_1} \mathcal{C}(\phi(\tau)^\star) \mathrm{d}\tau \leq \int_0^{t_1} \mathcal{C}(\phi(\tau)) \mathrm{d}\tau \tag{4.55}$$

Which, adopting the cost function 4.51, we can integrate analytically

$$\int_0^{t_1} \mathcal{C}(\phi(\tau)) \mathrm{d}\tau = - \int_0^{t_1} \sum_{i=0}^{n} \frac{1}{I_i(\tau)} \dot{I}_i(\tau) \mathrm{d}\tau = \sum_{i=0}^{n} (\log I_i|_0 - \log I_i|_{t_1}) \tag{4.56}$$

Then, comparing the integral of cost for the instantaneously optimum strategy $\phi^\star$ and all the other feasible ϕ we have that $\phi^\star$ maximizes the product of leftover resources

$$\log \left(\prod_{i=0}^{n} I_i^\star(t_1) \right) \geq \log \left(\prod_{i=0}^{n} I_i(t_1) \right) \tag{4.57}$$

This holds for every $t < t_f$. Note that, if at $t = t_1$ there exists a reachable point $\vec{I}(t_1)^\star$ strictly greater than all other feasible $\vec{I}(t_1)$, this is the best endpoint and the cost minimization strategy will reach it. The we would have consumed strictly less resources than in all other cases. This is the dual of maximizing responsive time; either we see it as consuming all resources and thus protracting operations as long as possible or as reaching a given time using as little resources as possible.

Final remarks

We have examined some specific cases within the linear model for **B**, **E**. We have found that, for a general request $\vec{R}(t)$, if we reach the origin we have maximized responsive time. Following this result, we have proposed a cost function that intuitively leads the system to the origin and we have shown that, when possible, this choice strictly minimizes resource consumption.

A number of issues remain open; some of them are listed here

- Even within the strict assumption of the linear model, we were unable to provided any strong locality results. While reaching the origin guarantees maximum responsive time, among two options at different *distances* from the origin, it is generally not true that the closer on will guarantee longer responsive time.
- The cost function proposed comes with no strong assurance; only weak behavior has been examined.

Chapter 5

Analytical System Design Using Cluster Scale Laws

Clusters are very versatile; they can be scaled quickly without extensive redesign and with high-confidence in both the aggregate performances (like total throughput) and the aggregate specifications (like total power consumption, volume etc). Furthermore, in the previous chapter we have shown how using decentralized algorithms can limit the computational overhead otherwise needed to control populous clusters.

We set out to exploit these features during the design phase. One option is to simply use Multi disciplinary Design Optimization and let the number of agents be a design variable. We will see how this already improves some significant drawbacks of MDO. However, using clusters we can also implement a novel design algorithm which, at the cost of some analytical assumptions, offers stronger results than standard MDO techniques.

We begin with the state of the art.

5.1 Literature review on Multi Disciplinary Optimization

System engineering problems usually feature the interaction of multiple different disciplines. The most reliable option to faithfully and reliably consider all couplings has traditionally been, and continues to be, the use of physical prototypes and test models. However, as computational simulations became more accurate, numerical techniques to simultaneously consider multiple discipline were developed. Furthermore, their relative low cost compared to the use of prototypes has enabled a shift in focus, from design validation to design optimization.

The prime driver of MDO has traditionally been the aerospace industry, where even a merely feasible design requires careful consideration of structural and aerodynamic trade offs. Historically, the origins of MDO can be seen in the works of Schmit [1],[2],[3] and Hafka [4],[5],[6], where optimization tecniques developed for structural design were extended to include other disciplines. Aircraft wing optimization encompassing aerodynamics, structure and control was initially pursued in [7],[8],[9],[10].

As a discipline in its own right, MDO is defined by the task of finding the values of the design variables which minimize an objective function under some constraints. In this respect, all MDO frameworks solve the same constrained optimization problem while providing different perks and drawbacks. Before presenting the specific merits of each, we briefly establish the nomenclature of the problem.

- The objective function is a map that associates to every design a scalar value; conventionally we want to find the design with the smallest possible value, so the objective function can be viewed as a penalty or cost function. Clearly, the choice of this function has a huge impact on the outcome of the optimization.

- The design variables uniquely identify a design; they are the independent variables set by the optimizer. For example, they could be the thickness of various structural elements, the wingspan, the chord of the wing profile etc.
  Design variables might interact with multiple disciplines, possibly with opposing effects on the objective function; increasing the thickness of the wings might be useful to reduce structural mass but might increase drag.

- The state variables are the output of the discipline specific analysis and hence can be naturally interpreted as dependent variables. State variables are directly determined by the choice of the design variables through dedicated analysis. For example, stress distribution in a loaded beam, lift distribution on an airfoil or temperature profile inside a combustion chamber are all outputs of domain specific analysis. However, as each analysis might depend on the state variables of other disciplines, some MDO architecture employ target states to allow parallelization. Each discipline then uses other disciplines target values to compute its own states, therefore avoiding an otherwise sequential process. The values for target states are chosen by the optimizer and additional consistency constraints need to be added to ensure that, within a few iterations, the actual state matches the target state.

- Constraints are functions of both design and state variable and are often strongly suggested by the problem. For example, lift is typically constrained to be equal to the mass of the vehicle; stress in a beam should be less than the elastic limit of the material etc.

We remark the fact that MDOs solve a numerical problem; while this dramatically reduces the cost of testing a design, it relies entirely on the problem formulation. While design variables and constraints naturally arise form the problem, the objective function is left to the best judgment of the system engineer.

MDO architectures differ in the ways in which they reach the optima. Although the universally attractive feature is to use the smallest number of evaluations possible, other criteria can also be considered. For example, it might be worthwhile to choose an architecture that allows early stop so that, even if the optimality condition has not been reached, the last iteration still satisfies requirements and is considerably better than the starting point. An other important feature might be the capability of using state of the art, black box tools without excessive code customization and software integration. Finally, more operational concern might also be addressed as one could prefer a solution which can be directly deployed on distributed computing architectures to reduce overall design time.

We now provide an overview of the most widely studied monolithic architectures. Any review of architectures for distributed computing, which break down and distribute the initial problem into smaller optimization tasks to be run on different computational nodes, is outside the scope of this work. Though not negligible, the main advantage of distributed MDO is to improve computational performance, and as they can all be traced back to some of the basic monolithic architecture, they do not contribute significantly to the main argument.

All At Once Problem Statement (AAO) AAO formulation [11] (refereed as Simultaneous Analysis and Design in [15],[12]) is the most general formulation of the problem. Everything is directly accessible by the optimizer, which sets design variables, target states and even state variables (only to then check the consistency constraints and satisfy them by choosing the latter two to be equal). All the equations of the various disciplines are directly available in residual form and appear as constraints.
This formulation has no practical use besides presenting the most general problem purely from the mathematical point of view. It is the starting point from which all other methods can be derived.

Simultaneous Analysis and Design (SAND) Eliminating the consistency constraints from AAO yields the SAND architecture [12]. The optimizer sets design variables and target states and tries to satisfies all constraints, including all discipline specific equations.
It has the advantage of being potentially faster, since between iteration there is no need for respecting feasibility. However, it considers a still very large problem (since each discipline is considered together with all others) and it requires direct access to all discipline equations. This is a significant drawback as most commercial software do not expose the mathematical implementation and are designed to output the state variables as a response to the design variables.
Hence, although it might be faster, SAND requires a lot of software customization.

Individual Discipline Feasible (IDF) IDF is also know as distributed analysis optimization [13] and optimizer based decomposition [14]. Here, the implicit function theorem is used to remove discipline analysis from the problem statement. The optimizer sets design variables and target states, lets the discipline specific codes provide the true value of the state variables and accounts for the discrepancy with consistency constraints.
This allows the use of commercially available, black box code to be used by the optimizer. The main drawback is that the problem is still very large compared to MDF, and thus might be slower. Also, when using a gradient based method, computing derivatives numerically is going to be very expensive, as each discipline specific simulation needs to converge for every point needed to estimate the gradient.

Multidisciplinary Feasible (MDF) This architecture is also called Fully Integrated Optimization [13] or Nested Analysis and Design [15]. It stems from removing both analysis and consistency constraints from the AAO architecture.
The optimizer has control only over the design variables and then runs every simulation in a loop until the state variables shared among various disciplines converge. Then it takes a single optimization steps and repeats.
The advantage is that the optimization problem is as small as it can be; furthermore it always returns a design configuration which is consistent with respect to the state variables. However, the problematic feature of IDF, namely the computational effort needed to compute the gradient, is only exacerbated in this case, as convergence among all state variables is needed for every evaluation.

As it transpires even from this short review, MDO research is focused on perfecting the solution of a single, albeit very complex and general, constrained optimization problem. Regardless of the chosen method, we point to three main problems;

1. Manufacturing and economic analysis require a very detailed design to reach an accuracy comparable with physic based simulations. Without using clusters, predicting manufacturing and economic figures is either extremely time consuming or necessarily low fidelity.

   Numerical tools for physical problems are very mature; answering manufacturing question with the same accuracy is harder. This situation is partially induced by the typical use of MDO, which is to facilitate the development of new concepts during the initial design phase, and thus manufacturing issues are seldom considered. On the other hand, to simulate manufacturing processes one would need a very detailed design of the actual components as well as the chosen manufacturing method, which often depends on external factors like available machinery or third party collaborations. The whole process thus requires a lot of additional inputs which would otherwise be sorted out much later, before the AIT phase.
   Hence, while prediction about physical parameters might be very accurate, MDO often takes a leap of faith with regard to manufacturability and implementation details.

   This drawback can be significantly reduced using clusters, as they provide assurance that the basic components already exist and thus are obtainable at a known cost and lead time, without the need for complex and dedicated simulations.

2. The choice of the objective function is subjective, yet optimization hinges upon it. We consider two complementary cases: either the objective is not based on a physical quantity, or it is.

   In the first case, we are implicitly referring to an economic objective, a *dollar-value* cost. As we mentioned before, without using cluster of components, cost predictions are typically not very accurate. Thus, by choosing an *economic* objective function we are relying on a loose indication at best, and a misleading model at worst.

   Optimizing for a physical characteristics is a more sound approach. For example, we might want to maximize some performance (say Isp), and possess accurate simulation tools that allows it. However, the true objective is rarely to obtain a high performance *per se*. More often it has to do with the effects that a high performance will have, such as reducing total fuel requirement and thus lunch cost, lowering operational cost etc. The idea being that, since the true objective can not be easily measured, a physical performance is used as a proxy for it. Following this strategy we might discover that, to increase a performance we could measure, we have sacrificed one that we could not, like maintenance complexity, manufacturing costs, software development cost etc.
   Paradoxically, while we turn to MDO because we recognize that the problem features too many complex interactions for a designer to handle, we still ask the engineer to anticipate all the side effects that the blind optimization of one specific parameter might have an all other disciplines.

3. When using large numerical models, it is very hard to guarantee that the solution is a global optima. In all cases beyond toy problems, the design space is very large and it becomes impossible to assure convergence to a global optima. Usually, it is accepted that the output of the optimization is some kind of local optima.

If we are to be very critical of MDO, we could observe that it uses a model which accurately captures only half the problem, while pursuing an objective that may or may not be what we want, with no guarantees that the output will actually optimize it. On the other hand, engineers must be practical in solving the problems at hand and MDO has been used successfully. The shear number of examples applied to complete aircraft [16],[17], bridges [18], buildings [19],[20],railway cars [21],[22], microscopes [23], automotive [24],[25], ships [26],[27], rotor craft [28],[29] and spacecraft [30],[31] proves that, in a very practical sense, MDO works.

However, the fact that a solution found by an MDO was manufactured successfully does not imply that the manufactured design was optimal. The concept of optimum is a feature of the mathematical model used to capture the problem and may or may not carry over onto the real system.

Therefore in this chapter we develop a design framework which shifts the emphasis from the optimization to the pursue of more robust guarantees; both in mathematical terms and in practical terms. Exploiting cluster properties and some additional analytical assumptions, we use the system architecture itself to guide the design and at the same time, improve the confidence that the resulting design specifications will be achievable.

A word to the wise

Formal results require formal proofs. This chapter casts system design into an unusually rigorous light. This will narrow the scope of the results we present; we will lay out the definitions with autonomous systems in mind. Particular attention will be given to small satellites, but most ideas and definitions should work also for land rovers, UAV and for general mobile systems which require a high degree of autonomy. Our objective is to show that, in some architectures and particularly when using clusters, there exists an optimization strategy that is cost agnostic.

The first sections are quite general, because the main result, (which is Lemma 9) allows them to be. However, to prove the hypothesis of the Lemma, we will make more specific assumptions. A case in which these assumptions are met is the subject of the next chapter, where the model is applied to a CubeSat and its subsystems.

5.2 Formal general idea

Our first goal is to obtain a system design that satisfies the mission objective.

We begin with a model of the physics of the problem which we can use to determine if a given system would be able to meet mission objective. This model will depend on parameters which describe the aggregate properties of the system at a high level, like total mass, average power consumption etc. These quantities will be known with confidence only after the design is completed, so at this stage, we can only guess them based on some heuristics, or our best judgment.

Upon finding a suitable combinations, we can claim that, if we were able to build a system with those parameters, we would be able to achieve the mission objective.

To test the conditional we increase the model resolution with regard to the system. From mission objective and the aggregate parameters we have hypothesized, we derive subsystem requirements. The subsystems are then designed to meet these requirements, and from their design specifications we can obtain more reliable estimates of the aggregate parameters.

We can therefore check if the assumptions that we used to assess mission feasibility were correct or not and either confirm the solution found, or start a new design cycle to examine a different set of parameters. This process is depicted in Fig 5.1.

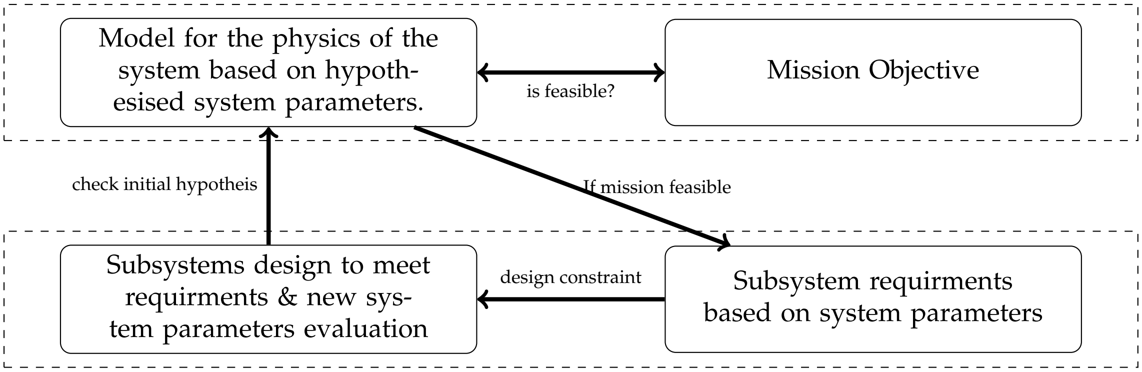

Model for the physics of the system based on hypothesised system parameters.
is feasible?
Mission Objective
check initial hypotheis
If mission feasible
Subsystems design to meet requirments & new system parameters evaluation
design constraint
Subsystem requirments based on system parameters

Figure 5.1: An abstraction model of the design process. A set of system parameter are assumed and tested against mission objective. If the design results to be feasible, the initial hypothesis of system parameters are checked against subsystem technological capabilities.

Generally, we might not have the luxury to perform the actual design of all subsystems during the exploration phase. Instead, we might resort to some performance trends, where significant specifications are plotted as a function of requirements (such as an energy density trends for batteries, or a typical bit rate for a type of radio).

After confirming our initial guesses with such curves, we might wonder how accurate the results we got were. To answer this question, we can apply the same procedure at a lower level; assigning requirements to the components that populates each subsystem, consulting the appropriate technological curves and so on.

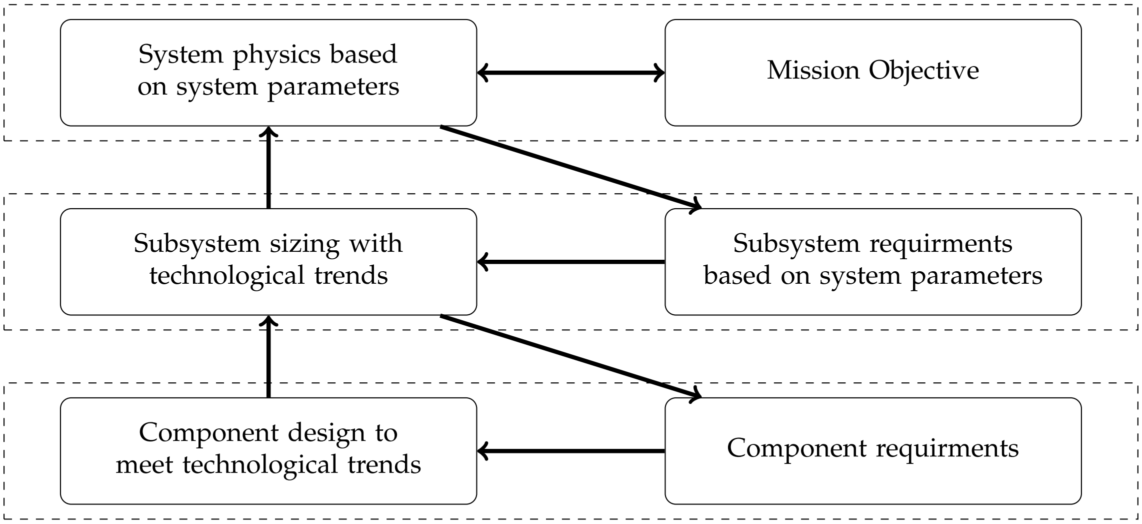

System physics based on system parameters
Mission Objective
Subsystem sizing with technological trends
Subsystem requirments based on system parameters
Component design to meet technological trends
Component requirments

Figure 5.2: A possible hirachy of system design; each level confirms or refutes the hypothesis made at the level above.

As we push deeper in the system hierarchy, we are adding information to the design by specifying the architecture to ever greater detail and confidence in the aggregate parameters increases. Eventually we end up with the complete system definition, where we know exactly which components we are using.

What we have described is a complete design cycle, which takes a considerable amount of resources. It would be unpractical to perform it completely during the exploration phase, therefore a compromise between the effort to test the solution and confidence in the parameters used has to be made. This is especially true for highly customized missions, like scientific ones, where the high degree of innovation renders predictions based on technological trends unreliable and therefore the system hierarchy needs to be developed on many layers.

In some cases however, the hierarchy can be quite shallow; the last iteration of the procedure is reached when we use COTS. If the subsystems are already COTS, their characteristics are well defined and we can have a high degree of confidence in their value. Similarly, if the subsystems are clusters of COTS, their scaling properties are well behaved, and we can have similar levels of confidence in the result.

The core idea is to express each step of the iterative procedure above in formal terms. This will enable us to make formal assertions on the system engineering process.

Overview of the algorithm So far, we have been concerned exclusively with finding a design/set of requirements that is

- **A**. Capable to accomplish mission objective

- **B**. Technically feasible

Reaching objective **A** will yield a system design that is mission worthy; **B** enforces the boundaries of current technological capability. There is no reason why conditions **A** and **B** would guarantee existence of a solution; indeed one can easily think of countless missions which are not physically or technologically feasible. If a solution exists however, there is no reason for it to be unique. Therefore traditionally, we add a final condition to the problem statement; *find a design, that* ***A***, ***B*** *and*

- **C**. Optimize a cost function $\mathcal{C}$.

Remark. Objectives **A** and **B** can also be viewed as the *characteristic condition*[1] which identifies the membership of an element of the design space to the sets $\mathcal{A}$ and $\mathcal{B}$ respectively. From the set $\mathcal{A}$, of all the systems which would satisfy the mission objective, we select only the ones we are able to actually produce ($\mathcal{B}$). Then we further refine our choice by selecting the best option according to some cost function $\mathcal{C}$.

[1] Also known in mathematics as indicator function or a characteristic function.

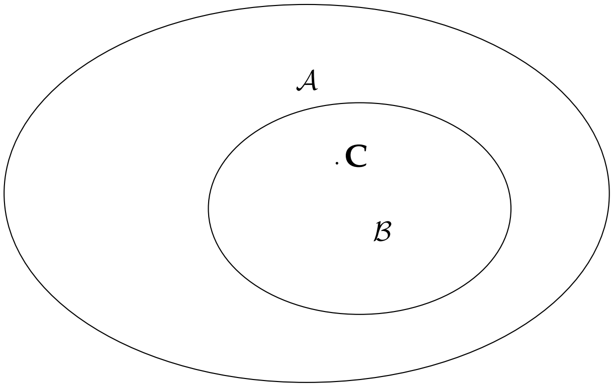


Figure 5.3: Condition **A,B** identify sets $\mathcal{A}, \mathcal{B}$, while $\mathcal{C}$ is a scalar function defined on $\mathcal{B}$, and minimized at the shown point.

5.2.1 The mission worthy set $\mathcal{A}$

At this point, we have:

- The mission definition
- A model for the system behavior/system physics

We view the system as an object which produces *signals* to achieve the mission objective. Using the physical model, we can simulate the mission and thus we can examine how the chosen design performs. Based on the performance of the simulated system we decide if the design is *mission feasible* or not. For example, if the mission requirement is to maintain the internal temperature within a given range, we simulate the mission to observe the behavior of the thermal control loop. During the simulation the system will react to environmental conditions, which are determined from both the mission and the system parameters, and apply some signals to achieve the mission objective.

To simulate the mission and, indirectly, to derive system actuations, we need to properly define the system design. We will call these system-level values, system parameters.

We can identify the **mission worthy set** as the set of all the system parameters for which there exists a set of actuation/output signals able to satisfy system requirements. We can think of system parameters as a point in the design space.

In mathematical terms, calling $\Phi_{\vec{m}}$ the model that we have chosen to reproduce the physics of a design with system parameters $\vec{m}$, the function $\Phi_{\vec{m}} : \vec{x}(t) \rightarrow \vec{r}'(t)$ maps the effects of subsystem actuations ,$\vec{x}(t)$, in the space of mission requirements, $\vec{r}'(t)$. The existence and knowledge of an inverse of $\Phi_{\vec{m}}()$ is the characteristic condition of the mission worthy set $\mathcal{A}$.

Note that the model for the physics $\Phi_{\vec{m}}$ can become arbitrarily complex. While the direct problem remains tractable and commercially available numerical tools can often be used to simulate it, the inverse is typically harder and requires dedicated analysis. For example, if the spacecraft thermal behavior is modeled with a single node, both the direct simulation and a rule to control its temperature can be implemented very easily. On the other hand, if we model the spacecraft with a fine mesh, we can still comfortably simulate the temperature distribution, but how to apply the heat to obtain a predefined temperature distribution is much harder. Since we need an inverse of $\Phi_{\vec{m}}$, in the design phase we are typically bounded to a simplified simulation. More detailed models may be be used in verification.

Therefore, the formal definition of the mission worthy set on the space of system parameters $\vec{m}$ is

$$\mathcal{A}_{\vec{r}(t)} = \left\{ \vec{m} \in \mathbb{R}^m \text{ for which } \exists^+ \mathbf{A}_{\vec{m}} \text{ such that } \Phi_{\vec{m}}(\mathbf{A}_{\vec{m}}(\vec{r}(t)) = \vec{r}(t) \right\} \tag{5.1}$$

It is worth noting that, for engineering purposes, existence $\mathbf{A}_{\vec{m}}(\cdot)$ is not enough. We need *knowledge* of the function $\mathbf{A}_{\vec{m}}(\cdot)$. We must have some kind of inverse function for $\Phi_{\vec{m}}$. If this was not the case, we would have no way to instruct the system on how to act autonomously to reach a desired output. We note that in general, it is not necessary for Φ_m to have a unique inverse. If the architecture is able to meet requirement in different ways, one can add optimization functions to close the problem.

5.2.2 The technologically worthy set $\mathcal{B}$

From the previous section, every design in $\mathcal{A}$ has an actuation $\vec{x}(t)$ able to meet mission requirements; yet it is not clear if such design is *technologically feasible*. Is this design within the reach of our technological capabilities?

The actuations computed using $\mathbf{A}_{\vec{m}}(\vec{r}_{\text{Mission}})$ are based on a set of system parameters, which we have identified as $\vec{m}$[2]. For the design to be *technologically feasible*, it will need to be able to deliver the outputs $\vec{x}(t)$ while not exceeding the system parameters used to compute them. We need to enforce a consistency constraint. For example, while any force can be provided given enough thrusters in parallel, acceleration is always technologically bounded by the thrust-to-mass ratio of the engine.

Given system requirements, we need to validate the initial assumptions on target system parameters. However, the concept of system is only a convenient abstraction for the collection of subsystems, each tasked with the production of different signals. Then, all the signals $\vec{x}(t)$ which the system must produce will be addressed by some internal component. After we design each subsystem to meet its respective requirement we can finally estimate system parameters.

To test internal consistency we need to design the subsystems to meet target actuation $\vec{x}_{\text{Mission}}(t)$. For example, the authority of each subsystem must be greater than the maximum output requested of it. Similarly, we need to have enough resources to sustain the output for the whole duration of the mission. It is easy to express these conditions formally. Calling X_i the authority of the subsystem i, t_{end} the time at which the mission ends

$$\mathcal{T} = [t_0, t_{\text{end}}] \qquad X_i \geq \max_{t \in \mathcal{T}} x_i(t) \tag{5.2}$$

For subsystems which use a consumable resource, we can impose conditions also on the integral of the output

$$X_{i+1} \geq \int_{t=0}^{t_{\text{end}}} x_i(t) \mathrm{d}t \tag{5.3}$$

We will refer to these X_i as subsystem requirements, which must be met by the respective subsystems. To handle them collectively with ease, we define a function $\mathbf{B}_1$ which extract the minimum

[2] Recall that these might include total mass, average specific heat,moment of inertia and so on.

subsystem requirements needed to provide a specific actuation $\vec{x}(t)$

$$\mathbf{B}_1 : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^m \qquad \mathbf{B}_1(\vec{x}(t)) = \vec{X} \tag{5.4}$$

In general, we can imagine $\mathbf{B}_1$ as a collection of functions similar to Eqs.5.2-5.3. Specific applications will justify the definition of specific components of $\mathbf{B}_1$; but initially it will be easier to consider only conditions as 5.2-5.3.

Having defined a set of requirements for each subsystem, we need to define a procedure that

1. provides a design solution for the subsystem that meets requirements

2. measures the subsystem contribution to system parameters $\vec{m}_2$ for the proposed solution

The two steps are thought as sequential; first design and then assessment of mass/volume contributions. However they will appear as a single step, which is what happens if performance curves are used.

$$\mathbf{B}_2 : \mathbb{R}^m \to \mathbb{R}^k \qquad \mathbf{B}_2(\vec{X}) \doteq \sum_{i=1}^{n} \mathbf{B}_{2,i}(\vec{X}) = \sum_{i=1}^{n} \vec{m}_i = \vec{m} \tag{5.5}$$

Function $\mathbf{B}_2$ contains how we design the subsystems in response to requirements $\vec{X}$ and all the contribution to the system's parameters. In general, it might be very hard to define this function on theoretical grounds. Typically it would be some combination of technological curves based on statistical data, however this approach fails when more custom designs are required. In the case of clusters of COTS, this is considerably easier and more reliable, as we can simply increase the number of units in the cluster until requirements are met.

Finally we can formally test the internal consistency constraint. A design is technologically feasible if its system parameters are compatible with the ones assumed to compute them.

$$\vec{m}_1 \text{ is feasible} \quad \Leftrightarrow \quad \vec{m}_1 \underset{=}{\ll} \mathbf{B}_2 \left\{ \mathbf{B}_1 \left[\mathbf{A}_{\vec{m}_1}(\vec{r}(t)) \right] \right\} \tag{5.6}$$

In other words, if we are able to produce a system capable of doing what it is requested of it, without exceeding any of the budgets that were assumed to compute its requirements, it is technologically feasible. In figure 5.4, we summarize this formal version of the engineering process

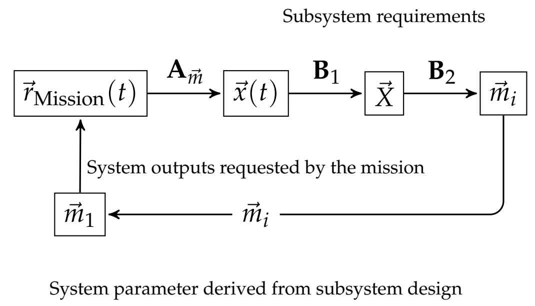


Figure 5.4: A diagram of the conditions to check for consistency at the subsystem level.

Having defined the characteristic function for the set $\mathcal{B}$, we would like to improve the set characterization presented in Fig 5.3. The previous definition for the sets $\mathcal{A}, \mathcal{B}$ were

$$\mathcal{A} \doteq \left\{\vec{m} \in \mathbb{R}^m : \exists^+ \mathbf{A}_{\vec{m}} \text{ such that } \Phi_{\vec{m}}(\mathbf{A}_{\vec{m}}(\vec{r}(t)) = \vec{r}(t)\right\} \tag{5.7}$$

$$\mathcal{B} \doteq \left\{\vec{m} \in \mathbb{R}^m \text{ such that } \exists^+ \mathbf{A}_{\vec{m}} \text{ and } \mathbf{B}_2\left(\mathbf{B}_1\left(\mathbf{A}_{\vec{m}_1}(\vec{r}(t))\right)\right) \underset{=}{\ll} \vec{m}_1\right\} \tag{5.8}$$

Which remarks $\mathcal{B} \subset \mathcal{A}$.

Remark. This representation of $\mathcal{B}$ puts the stress on the function $\mathbf{B}_2$ which is the most important component of system design. However, there can be different yet equivalent definitions of set $\mathcal{B}$. In the proofs it will be more intuitive to refer to the element of $\mathcal{B}$ as vector of subsystem requirements. Having identified the function $\mathbf{B}_1$ and $\mathbf{B}_2$, we can map $\vec{m}$ to a respective $\vec{X}$ as

$$\vec{X} = f(\vec{m}) \quad \text{by choosing} \quad \vec{X} \,|\, \mathbf{B}_2(\vec{X}) = \vec{m} \tag{5.9}$$

The definition of $\mathcal{B}$ becomes

$$\mathcal{B} \doteq \left\{\vec{X} \text{ such that } \mathbf{B}_2\left(\mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X})}(\vec{r}(t))\right)\right) \underset{=}{\ll} \mathbf{B}_2(\vec{X})\right\} \tag{5.10}$$

Which appears quite more convoluted, but is equally valid.

5.2.3 Minimization of cost function

The cost function has been presented as an expedient to automate the choice of a design among those which are feasible. Therefore, its domain is defined as the technologically feasible set $\mathcal{B}$, while its co-domain must possess a strong ordering so it must be a subset of $\mathbb{R}$.

It seems more intuitive to the author to define the cost function as a function of both system parameter and subsystem requirements. However, as it is possible to represent the elements of $\mathcal{B}$ in many equivalent ways, this choice does not affect the generality of the argument. We operationally define the cost function $\mathcal{C}$ as

$$\mathcal{C}(\vec{X}, \vec{m}) \to \mathbb{R} \tag{5.11}$$

so that we can directly assign a cost contribution to both subsystem requirements (such as authority, capacity, bandwidth etc) and to any the aggregate system parameter.

The main assumption we make is that $\mathcal{C}()$ is a reasonable cost function if any increase in subsystem requirements will not decrease cost. Formally, we translate this idea into the condition for $\mathcal{C} : \mathcal{B} \to \mathbb{R}$ to be *smooth* and monotone with respect to all subsystem requirements, hence

$$\frac{\partial \mathcal{C}}{\partial X_i} \geq 0 \quad \forall i \tag{5.12}$$

We expect that a system composed of subsystems with high performances will not cost less than one which uses subsystems with lower performances. An EPS design capable of harvesting more solar power will not cost less than one which is able to produce less power. This appeals to a very economic idea of *cost*, and it is easy to accept. After all, were this not the case, we could simply use the *better* system whenever the *worse* system would be used.

However, we noted that $\vec{X}$ and $\vec{m}$ are not independent. Since we have introduced function $\mathbf{B}_2()$, we have established a correspondence between the subsystem requirements $\vec{X}$ and the impact that a subsystem will have on the system if it was designed to meet them, $\vec{m}$. Therefore using $\mathbf{B}_2$ we can write

$$\mathcal{C}(\vec{X},\vec{m}) = \mathcal{C}(\vec{X},\mathbf{B}_2(\vec{X})) \quad \Rightarrow \quad \mathcal{C} = f(\vec{X}) \tag{5.13}$$

which allows us to examine the hypothesis 5.12 in more details

$$\frac{\partial\mathcal{C}}{\partial X_i} = \frac{\partial\mathcal{C}}{\partial X_i} + \sum_{j}^{n_m}\left(\frac{\partial\mathcal{C}}{\partial m_j}\cdot\frac{\partial\mathcal{B}_{2,j}}{\partial X_i}\right) \geq 0 \tag{5.14}$$

The significance and interpretation of the first term remains the same. The second terms however addresses the indirect effects of an increase in any requirement. The term $\partial\mathcal{C}/\partial m_j$ states whether it is preferable to increase or decrease a specific system parameter. The $\partial\mathcal{B}_{2,j}/\partial X_i$ term states if an increase in a system requirement increases or decreases system parameters.

It might be challenging to defend this hypothesis on general grounds. With a specific application in mind however, these assumptions can be quickly accepted or refuted. In the case of a small satellite, we can show that these are not too extravagant. While a more detailed analysis is given in the next chapter, we can provide hints of why this might be the case. We consider a small satellite in which the relevant subsystems are on board computer, EPS, ADCS, telecommunication and payload. The respective requirements might be stated in terms of processor clock frequency, orbit average power, pointing accuracy and bit rate. It is easy to accept that any faster processor, more accurate ADCS or more powerful radio will not cost less than a less capable counterpart. Moreover, if we consider secondary effects we can think that an increase in radio output power will increase demands on the power budget (if the efficiency is kept constant). This will have a negative effect on the EPS, which will need to be more capable. On the other hand, if we were to increase transmission efficiency by using a bigger antenna for example, we would avoid to overload the EPS but we would negatively impact the ADCS due to an increase in inertia.

Of course, this is just a model and, as such, we should expect it to be a significant simplification of reality. There might be cases in which an increase in one requirement actually does decrease cost. However, the usefulness of this hypothesis is that it is rather safe and needs to be verified on average, due to the sum of the partial derivative contributions.

From now on, when referring to a cost function, we will assume that hypothesis 5.12 is satisfied.

5.3 Existence of a strong optimum

It is possible, in some cases, to avoid choosing a cost function. If we know about the existence of a point which minimized all cost functions at once, it is not necessary to provide a correct one.

Using the notation introduced in the previous section, where $\mathcal{B}$ is the set of designs identified by their requirements $(\vec{X})$, we can state the following Lemma

Lemma 9. Let $\mathcal{F}(\cdot)$ be a contraction mapping over the complete metric space of design defined as

$$\begin{array}{rccc} \mathcal{F}: & \mathcal{B} & \rightarrow & \mathcal{B} \\ & \vec{X} & \rightarrow & \mathcal{F}(\vec{X}) \end{array} \qquad \mathcal{F}(\vec{X}) \doteq \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X})}(\vec{r}(t))\right) \tag{5.15}$$

And

$$\vec{X}_1 \underset{=}{\ll} \vec{X}_2 \quad \Rightarrow \quad \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_1)}(\vec{r}(t))\right) \underset{=}{\ll} \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_2)}(\vec{r}(t)\right) \tag{5.16}$$

Then there exists a unique fixed point $\vec{X}^\star$ for $\mathcal{F}(\cdot)$ and it minimizes any cost function $\mathcal{C}(\cdot)$ for which $\frac{\partial \mathcal{C}}{\partial X_i} > 0 \ \forall i$.

Proof. To begin with, if $\mathcal{F}$ is a contraction mapping, we know that the fixed point $\vec{X}^\star$ exists. Notice that if $\vec{X}^\star$ exists, we can prove that it must be in $\mathcal{B}$ so we know that $\mathcal{B} \neq \emptyset$. This is shown by the following

$$\exists \vec{X}^\star \text{ such that } \mathcal{F}(\vec{X}^\star) = \vec{X}^\star \quad \Rightarrow \quad \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}^\star)}(\vec{r}(t))\right) = \vec{X}^\star \quad \Rightarrow$$

$$\Rightarrow \quad \mathbf{B}_2\left[\mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}^\star)}(\vec{r}(t))\right)\right] = \mathbf{B}_2(\vec{X}^\star) \quad \Rightarrow \quad \underbrace{\mathbf{B}_2\left\{\mathbf{B}_1\left[\mathbf{A}_{\vec{m}^\star}(\vec{r}(t))\right]\right\} \underset{=}{\ll} \vec{m}^\star}_{\text{characteristic condition for}\mathcal{B}} \quad \Rightarrow \vec{X}^\star \in \mathcal{B}$$

For any cost function $\mathcal{C}(\cdot)$ we have that

$$\forall \vec{X}_1, \vec{X}_2 \in \mathbb{R}^{n+} \qquad \vec{X}_1 \underset{=}{\ll} \vec{X}_2 \Rightarrow \mathcal{C}(\vec{X}_1) \leq \mathcal{C}(\vec{X}_2)$$

Then, if there exists a point $\vec{X}^\star \in \mathcal{B}$ such that $\vec{X}^\star \underset{=}{\ll} \vec{X} \quad \forall \vec{X} \in \mathcal{B}$ it will be a minimum for any cost function $\mathcal{C}(\cdot)$, as

$$\vec{X}^\star \underset{=}{\ll} \vec{X} \quad \Rightarrow \quad \mathcal{C}(\vec{X}^\star) \leq \mathcal{C}(\vec{X}) \qquad \forall \vec{X} \in \mathcal{B}$$

We will prove that such $\vec{X}^\star$ is the fixed point for $\mathcal{F}$.

Again by the contraction mapping theorem, we know that $X^\star$ exists and is unique. Assume, by contradiction, that $X^\star$ is not a minimum for $(\mathcal{B}, \underset{=}{\ll})$, either because the minimum does not exists, or because there is a point which is strictly smaller. This is equivalent to the existence of some $\vec{X}_2 \in \mathcal{B}$ such that

$$\vec{X}^\star \not\underset{=}{\ll} \vec{X}_2$$

Either the two points are not comparable or $\vec{X}_2$ is a possible candidate for the minimum and it is not a fixed point for $\mathcal{F}$. Then, we have that

$$\text{since} \quad \begin{cases} \vec{X}_2 & \in & \mathcal{B} \\ \vec{X}_2 & \neq & \vec{X}^\star \end{cases} \quad \Rightarrow \quad \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_2)}(\vec{r}(t))\right) \ll \vec{X}_2 \tag{5.17}$$

Notice that we can use a strict inequality since $\vec{X}_2$ is not a fixed point. Then we can define

$$\vec{X}_3 \doteq \mathcal{F}(\vec{X}_2) \ll \vec{X}_2$$

and, since $\mathbf{B}_1(\mathbf{A}_{\mathbf{B}_2(\vec{X})})$ is non decreasing with $\vec{X}$ by hypothesis 5.16, we know that

$$\mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_3)}(\vec{r}(t))\right) \ll \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_2)}(\vec{r}(t))\right) \quad \Rightarrow \quad \mathcal{F}(\vec{X}_3) \underset{=}{\ll} \vec{X}_3$$

Which means that $\vec{X}_3 \in \mathcal{B}$. The contraction mapping theorem tells us that, if we apply the contraction $\mathcal{F}$ from any point in the metric space, we will converge to the fixed point $\vec{X}^\star$. In this case however, we also know that if we apply $\mathcal{F}$ within $\mathcal{B}$ we remain within $\mathcal{B}$. Also, we are moving on a decreasing sequence (for any norm)

$$\vec{X}_{i+1} = \mathcal{F}(\vec{X}_i) = \mathbf{B}_1 \left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_i)}(\vec{r}(t)) \right) \ll \vec{X}_i \tag{5.18}$$

But, since the fixed point is unique, we will have to converge to the point $\vec{X}^\star$ with a chain of decreasing inequalities, for which

$$\vec{X}_2 \gg \vec{X}_3 \gg \vec{X}_4 \gg \vec{X}_5 \gg \cdots \gg \vec{X}_n = \vec{X}^\star \quad \Rightarrow \quad \vec{X}_2 \gg \vec{X}^\star$$

which is against hypothesis. Therefore a minimum for any norm/cost exists and it is the fixed point. □

This result may seem strange, and perhaps counter intuitive. Provided that the hypothesis of Lemma 9 are met, the existence of a system which is optimal for every cost function is at the very least suspicious. One key distinction to keep in mind is that the minimization of every cost function does not mean that the system is optimal for every task or mission. Quite the opposite; an accurate interpretation would be that, for some specific mission, there is only one optimal system.

Finally, note that the iteration procedure from the contraction mapping theorem provides a procedure to find the fixed point $\vec{X}^\star$.

Hypothesis in the Lemma

The hypothesis of Lemma 9 are very technical and, as presented, may be hard to assess. To test if they are true for some particular engineering application, it is easier to work with the specific models for the function $A_{\vec{m}}, \mathbf{B}_1, \mathbf{B}_2$. If generic proofs exist, they have escaped the author. The cases we present can be almost directly applied to small satellites, and CubeSats in particular.

The hypothesis 5.16 will be discussed here, while deriving a condition for $\mathcal{F}()$ to be a contraction map (hp 5.15) is a bit longer and will be developed in the next section.

To ask $\mathbf{B}_1 \left(\mathbf{A}_{\mathbf{B}_2(\vec{X})} \left(\vec{r}(t) \right) \right)$ non decreasing with $\vec{X}$ is to prove Eq.5.16

$$\forall \vec{X}_1 \underset{=}{\ll} \vec{X}_2 \in \mathcal{B} \quad \Rightarrow \quad \mathbf{B}_1 \left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_1)}(\vec{r}(t)) \right) \underset{=}{\ll} \mathbf{B}_1 \left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_2)}(\vec{r}(t) \right) \tag{5.19}$$

This fits well with the idea that, as long as we are able to meet mission requirements, there is nothing to be gained by over sizing a subsystem. In other words, more demanding subsystem requirements ($\vec{X}_2$) will not produce less demanding subsystem requirements. For example, requesting an engine with *more than necessary* authority will negatively impact the mass of the system hence will not decrease in the minimum required authority.

To begin with, we notice that if all functions preserve the component wise $\ll$ order, condition 5.16 is verified. This is to prove that

$$\begin{cases} \vec{X}_1 \ll \vec{X}_2 & \Rightarrow \quad \mathbf{B}_2(\vec{X}_1) \ll= \mathbf{B}_2(\vec{X}_2) \\ \vec{m}_1 \ll \vec{m}_2 & \Rightarrow \quad \mathbf{A}_{\vec{m}_1}(\vec{r}(t)) \ll= \mathbf{A}_{\vec{m}_2}(\vec{r}(t)) \\ \vec{x}_1(t) \ll \vec{x}_2(t) \quad t \in [0, t_{\text{end}}] & \Rightarrow \quad \mathbf{B}_1(\vec{x}_1(t)) \ll= \mathbf{B}_1(\vec{x}_2(t)) \end{cases} \tag{5.20}$$

which is a set of merely sufficient conditions. The first condition is about our technology, the second about the model we use to express the actuations and the third is about the tests we need to satisfy to produce a coherent system.

We proceed backward in the chain of conditions 5.20. The condition on $\mathbf{B}_1$ is trivially true as long as we only allow $\mathbf{B}_{1,i}(x_i(t))$ to be max $x_i(t)$[3].

The second part hinges on the model we use to represent the physics of the system. The most simple model which is still useful is the linear one; we view system parameters as amplification coefficients

$$\begin{aligned} x_1(t) &= r_1(t) \cdot m_1 \\ x_2(t) &= r_2(t) \cdot m_2 \\ &\vdots \\ x_n(t) &= r_n(t) \cdot m_n \end{aligned} \tag{5.21}$$

Examples that fits this behavior are how mass amplifies acceleration into force, inertia acts similarly on angular acceleration and the inefficiency $1/\varepsilon$ of the power distribution system, amplifies the power requirement. With this model, it is very easy to verify that

$$\vec{m}_1 \ll \vec{m}_2 \quad \Rightarrow \quad \mathbf{A}_{\vec{m}_1}(\vec{r}(t)) = \begin{pmatrix} m_1^{(1)} \cdot r_1(t) \\ m_2^{(1)} \cdot r_2(t) \\ \vdots \\ m_n^{(1)} \cdot r_n(t) \end{pmatrix} \ll= \begin{pmatrix} m_1^{(2)} \cdot r_1(t) \\ m_2^{(2)} \cdot r_2(t) \\ \vdots \\ m_n^{(2)} \cdot r_n(t) \end{pmatrix} = \mathbf{A}_{\vec{m}_2}(\vec{r}(t)) \tag{5.22}$$

We point out that this is certainly not the only model which satisfies the condition above. Similar results can be obtain using a generic polynomial function of $r_i(t)$, such as

$$x_i(t) = m_1 r_1(t) + m_2 r_1^2(t) + \cdots + m_j r^j(t) \tag{5.23}$$

which is more general and allows for more complex model; however working with the linear model will be much easier.

Finally, we want to prove that more demanding subsystem requirements do not have a beneficial impact on system parameters

$$\vec{X}_1 \ll \vec{X}_2 \quad \Rightarrow \quad \mathbf{B}_2(\vec{X}_1) \ll= \mathbf{B}_2(\vec{X}_2)$$

The process that underlines $\mathbf{B}_2$ is undoubtedly very complex; it is the whole engineering endeavor that, starting from subsystem requirement $\vec{X}$, produces the system design and measures some meaningful system parameters $\vec{m}$. However, we only care about its external behavior; it is sufficient to show that, for any of the system parameters m_j which are the outputs of $\mathbf{B}_2$, it is true that

$$\frac{\partial m_j}{\partial X_i} \geq 0 \quad \forall i, j \tag{5.24}$$

This is enough to prove the first of 5.20. We model system parameters as the sum of the contributions of each subsystem to subsystem parameter. A subsystem might contributes to the total mass,

[3] Different models for $\mathbf{B}_{1,i}$ will be presented as they are needed, with the respective proofs

inertia, etc. This is motivated by the necessarily simple model that we are using but allows us to separate the contribution for each subsystem as

$$\mathbf{B}_2(\vec{X}) \doteq \sum_{i=1}^{n} \vec{m}_{\text{sys }=i}\,(\vec{X})$$

Then we can test, for each subsystem if each component of its contribution m_i is non decreasing with any of its relevant requirement X_i

$$\frac{\partial m_{sys=i,j}}{\partial X_k} \geq 0 \quad \forall i, j, k \tag{5.25}$$

We can also notice that it is not necessary for this condition to be true for all subsystems; it would be enough to have it be true collectively.

At this point, we have reached a practical understanding of what we are asking each subsystem or component; given more demanding requirements, the impact the subsystem has on the system should not decrease. A more authoritative system can not have smaller mass than a less authoritative one. This is rather reasonable, and one could prove this statement using statistical trends for the subsystem of interest for the mission, but there is a special case in which it is not needed.

We can note that a system with higher authority can do everything a system with less authority can. If an option with more authority would produce a strictly smaller contribution to $\vec{m}$, a smart designer would simply choose the more authoritative one with a redefinition of the technological curve as in depicted in Fig. 5.5.

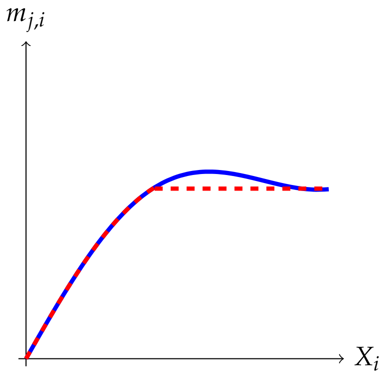


Figure 5.5: An example of redefinition of the technological curve, in red dashed.

However, there might be cases in which increasing the requirements would lead to a contribution $\vec{m}\,'$ which is not comparable with the contribution of a less performing component. This means

$$\vec{Y}, \vec{Z} \in \mathcal{B} \quad \vec{Y} \ll \vec{Z} \quad \begin{cases} B_2(\vec{Y}) & = & \vec{m}_y \\ B_2(\vec{Z}) & = & \vec{m}_z \end{cases} \quad \text{yet} \quad \begin{cases} \vec{m}_y & \not\ll & \vec{m}_z \\ \vec{m}_y & \not\gg & \vec{m}_z \end{cases} \tag{5.26}$$

This situation can not happen if we design the system using cluster of components, as both $\vec{m}$ and the requirement $\vec{X}$ are a linear function of the basic element properties. This means that we can specify the design of a subsystem by specifying the number of agents n in it based on the required authority

$$\vec{m}_i(X) = \begin{pmatrix} \text{mass} \\ \text{volume} \\ \text{power} \end{pmatrix} = \begin{pmatrix} m_1 \\ v_1 \\ p_1 \end{pmatrix} \cdot n \qquad n \text{ such that } \begin{cases} n \cdot T_{\max} & > \quad X_{T_{\text{req}}} \\ n \cdot H_{\max} & > \quad X_{H_{\text{req}}} \end{cases} \tag{5.27}$$

At least for clusters of components, this condition can be verified by design.

5.3.1 Model set up

The first complete system model which we will consider to prove $\mathcal{F}()$ to be a contraction map has the following representation

$$\mathbf{A}_{\vec{m}}(\vec{r}(t)) = \vec{x}(t) \qquad \begin{bmatrix} m_{1,1}(\vec{X}) & 0 & \dots & 0 \\ 0 & m_{2,2}(\vec{X}) & \dots & 0 \\ & \dots & \dots & \\ 0 & 0 & \dots & m_{n,n}(\vec{X}) \end{bmatrix} \cdot \begin{pmatrix} r_1(t) \\ r_2(t) \\ \dots \\ r_n(t) \end{pmatrix} = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_n(t) \end{pmatrix} \tag{5.28}$$

We will compact notation as $\mathbf{A}_{\mathbf{B}_2(\vec{X})}(\vec{r}(t)) = \mathrm{M}(\vec{X}) \cdot \vec{r}(t)$, hiding the middle passage through $\vec{m} = \mathbf{B}_2$.

The model for system parameters $\mathbf{B}_2$ will assume independent contributions from all subsystems ($\vec{m} = \sum_i^n \vec{m}_i$), while the individual functions $m_{i,i}(\vec{X})$ will be assumed to be non decreasing. We also require $\mathbf{B}_2$ to be Lipschitz continuous, which can be viewed as requiring that there exists a bound on the value of the derivative.

Note that the initial model, as shown in Eq.5.28 assumes independent actuations, meaning that actuators' output do not interact, or have any dependency relationship. A more realistic model would use

$$\begin{bmatrix} m_{1,1} & 0 & \dots & 0 \\ 0 & m_{2,2} & \dots & 0 \\ & \dots & \dots & \\ 0 & 0 & \dots & m_{n,n} \end{bmatrix} \cdot \begin{pmatrix} r_1(t) \\ r_2(t) \\ \dots \\ r_n(t) \end{pmatrix} = \begin{bmatrix} 1 & -\eta_{1,2} & \dots & -\eta_{1,n} \\ -\eta_{2,1} & 1 & \dots & -\eta_{2,n} \\ & \dots & \dots & \\ -\eta_{n,1} & -\eta_{n,2} & \dots & 1 \end{bmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_n(t) \end{pmatrix} \tag{5.29}$$

In which coupling between subsystem is assumed linear on average. Clearly Eq 5.28 is a special case of the general version above (Eq. 5.29). We will begin by proving the simple one and then expand towards the more general case.

For subsystem requirements we will start by considering only the actuator **authorities**, or maximum value of output over the time interval of the mission $\tau = [t_0, t_{\text{end}}]$. This means

$$\vec{X} = \mathbf{B}_1(\vec{x}(t)) = \begin{pmatrix} \max_{t\in\tau} x_1(t) \\ \max_{t\in\tau} x_2(t) \\ \dots \\ \max_{t\in\tau} x_n(t) \end{pmatrix} \qquad \text{where } x_i(t) \geq 0 \quad \forall i, \forall t \in \tau \tag{5.30}$$

The non negativity of the requested output x_i is to avoid the assumption of symmetry in the behavior of an actuator; the idea being that a negative sign would imply reversing the behavior, rendering an engine a generator for example. [4]. Later we will show that this model of $\mathbf{B}_1$ can be used to compute other types of requirements, like capacity.

5.3.2 An example toy problem

We consider a simple system, with only one engine and a thermal control actuator

$$\begin{pmatrix} a(t) \\ \dot{T}(t) \end{pmatrix} = \begin{bmatrix} 1/m_s & 0 \\ 0 & 1/m_t \end{bmatrix} \cdot \begin{pmatrix} F(t) \\ q(t) \end{pmatrix}$$

Requirements are given in terms of acceleration and maximum temperature gradient. m_s is the system mass and m_t is the thermal inertia of the system. The subsystem requirement space is a subset of $\mathbb{R}^2$, in which we look for a couple of authorities such that

$$\vec{A} = \begin{cases} \mathrm{Aut}_{Engine} & > & \max_t F(t) & = & \max_t a(t) \cdot m_s & = & a_{\max} \cdot m_s \\ \mathrm{Aut}_{Thermal} & > & \max_t q(t) & = & \max_t \dot{T}(t) \cdot m_t & = & \dot{T}_{\max} \cdot m_t \end{cases}$$

Where the system properties m_s and m_t are the sum of the subsystems contributions

$$m_s = m_{PL} + m_e + m_{tc} \qquad m_t = m_{t,PL} + m_{t,e} + m_{t,ts}$$

Finally, we recall the contraction map $\mathcal{F} : \mathbb{R}^2 \to \mathbb{R}^2$, which takes as input subsystem requirements and model the process of designing the subsystem to meet those specifications, extract system parameters from the proposed configuration, simulates the mission and finally updates subsystem requirements

$$\mathcal{F}(\vec{A}) = \begin{pmatrix} a_{max} \cdot m_s(\vec{A}) \\ \dot{T}_{max} \cdot m_t(\vec{A}) \end{pmatrix}$$

Due to the contraction mapping theorem, if we prove that, for a given norm $\|\cdot\|$ and within a set $\mathcal{S} \subset \mathbb{R}^2$

$$\|\mathcal{F}(\vec{A}_2) - \mathcal{F}(\vec{A}_1)\| < \|\vec{A}_2 - \vec{A}_1\|$$

then there is a unique fixed point $\vec{A}^\star$, which corresponds to the optimal system, and which can be reached by applying the contraction map recursively.
We only need to figure out under which assumption this is the case. Then

$$\left\| \begin{pmatrix} a_{max} \cdot (\cancel{m_{pl}} + m_e(F_2) + m_{tc}(Q_2) \cancel{- m_{pl}} - m_e(F_1) - m_{tc}(Q_1)) \\ \dot{T}_{max} \cdot (\cancel{m_{t,PL}} + m_{t,e}(F_2) + m_{t,ts}(Q_2) \cancel{- m_{t,PL}} - m_{t,e}(F_2) - m_{t,ts}(Q_1)) \end{pmatrix} \right\| =$$

$$= \left\| \begin{pmatrix} a_{max} \cdot (\Delta m_e(F) + \Delta m_{tc}(Q)) \\ \dot{T}_{max} \cdot (\Delta m_{t,e}(F) + \Delta m_{t,ts}(Q)) \end{pmatrix} \right\| < \left\| \begin{pmatrix} \Delta F \\ \Delta Q \end{pmatrix} \right\|$$

Using a homogeneous L_1 norm, we have

$$|a_{max} \cdot (\Delta m_e(F) + \Delta m_{tc}(Q))| + |\dot{T}_{max} \cdot (\Delta m_{t,e}(F) + \Delta m_{t,ts}(Q))| < |\Delta F| + |\Delta Q|$$

[4] In general, one should not assume it possible for an actuator to behave in a symmetric fashion. To model output which could be applied in different directions, we can simply add a virtual components which acts in the opposite direction

Clearly, a sufficient but very coarse condition for this to be true is that

$$\begin{cases} |a_{max}\cdot(\Delta m_e(F)+\Delta m_{tc}(Q))| & < & |\Delta F| \\ |\dot{T}_{max}\cdot(\Delta m_{t,e}(F)+\Delta m_{t,ts}(Q))| & < & |\Delta Q| \end{cases} \tag{5.31}$$

What is the meaning of Eq. (5.31) above? Momentarily, we imagine that the contribution that the engine makes to the thermal inertia is negligible, as the contribution that the thermal system makes on the overall mass. Then we obtain the trivial result

$$\begin{cases} a_{max}\cdot|\Delta m_e(F)| & < & \Delta F \\ \\ \dot{T}_{max}\cdot|\Delta m_{t,ts}(Q)| & < & \Delta Q \end{cases} \tag{5.32}$$

This means that we ask both the engine and the thermal control system technologies to be able to output more than the penalty they impose. A bigger engine must provide more force than the inertial force it requires at the maximum system acceleration a_{max}; in some sense, it must be worth it. It is important to remark that this condition is as much a prerogative of the chosen technology as it is of the mission requirement. The set of points of the authority space for which conditions 5.31 are verified is the set for which we know that an optimal solution exists, given the mission.

We now go back to considering all terms in the equation 5.31, which can be re-written as

$$\begin{cases} a_{max}\cdot\left(|\frac{\Delta m_e(F)}{\Delta F}|+|\frac{\Delta m_{tc}(Q)}{\Delta F}|\right) & < & 1 \\ \dot{T}_{max}\cdot\left(|\frac{\Delta m_{t,e}(F)}{\Delta Q}|+|\frac{\Delta m_{t,ts}(Q)}{\Delta Q}|\right) & < & 1 \end{cases} \tag{5.33}$$

The extended interpretation considers also the coupling between the subsystems; we are asking for the cumulative increase in requested authority for each subsystem to be smaller than the actual increase in authority for that subsystem. For example, the first condition of 5.33 bounds the increase in system mass, be it due to an increase of engine mass or increase in thermal control system mass to the increase of engine thrust. While an increase in engine mass is justified by an increase in engine output and therefore *feels* pertinent to the sizing of the engine itself, the coupling term is purely systemic feature.

The above model will be expanded in more formal terms in the next section.

5.3.3 Proof of $\mathcal{F}$ contraction map

Using the nomenclature and simplified models defined in the previous section, we want to provide sufficient conditions for both technology and the mission requirements that guarantee the existence of the Strong Optimum point. We need to prove that $\mathcal{F}()$ is a contraction mapping.

Lemma 10. Let $\vec{r}(t)$ be mission requirements, $R_i \doteq \max_t r_i(t)$ and $\mathbf{A}_{\vec{m}}, \mathbf{B}_1, \mathbf{B}_2$ as defined in 5.28,5.30 respectively. $\mathcal{F}()$ is a contraction mapping if there exists, $q \in [0,1)$ and $\hat{c} \in \mathbb{R}^{n++}$ such that:

$$\underbrace{\begin{bmatrix} \frac{\partial m_1}{\partial X_1}\cdot R_1 - q & \frac{\partial m_2}{\partial X_1}\cdot R_2 & \dots & \frac{\partial m_n}{\partial X_1}\cdot R_n \\ \frac{\partial m_1}{\partial X_2}\cdot R_1 & \frac{\partial m_2}{\partial X_2}\cdot R_2 - q & \dots & \frac{\partial m_n}{\partial X_2}\cdot R_n \\ \dots & \dots & \dots & \dots \\ \frac{\partial m_1}{\partial X_n}\cdot R_1 & \dots & \dots & \frac{\partial m_n}{\partial X_n}\cdot R_n - q \end{bmatrix}}_{\doteq \mathbf{Q}} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix} \ll= \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix} \qquad \mathbf{Q}\cdot\hat{c} \ll \vec{0} \tag{5.34}$$

Note that the above is only a sufficient condition; the proof is littered with coarse comparison for convenience.

Some remarks before the proof
We can immediately observe that there can not exists a suitable $\hat{c} \gg 0$ if any of the term on the diagonal (in Eq. 5.42) are positive. This provide some insights on the type of conditions that we are asking for $\mathcal{F}()$ to be a contraction mapping. Consider the limit case were $q = 1$. A necessary condition for $\vec{c}$ to exists is that for all $i = 1, \ldots, n$,

$$\frac{\partial m_i}{\partial x_i} R_i - 1 < 0 \quad \Rightarrow \quad \frac{\partial m_i}{\partial x_i} R_i < 1 \quad \Rightarrow R_i \int_0^{X_i} \frac{\partial m_i}{\partial x_i} \mathrm{d}x_i < \int_0^{X_i} 1 \mathrm{d}x_i$$

otherwise no $\hat{c} \gg 0$ could not produce a strictly negative vector. This means that

$$R_i \cdot (m_i(X_i) - m_i(0)) \leq X_i \tag{5.35}$$

Which bears immediate engineering meaning. Consider an engine; condition 5.35 requires the increment in force $R \cdot \Delta m$, caused by the increment in required authority of the engine, to be smaller than the increment of force achieved. If this was not the case, increasing the authority of the engine would decrease our ability to accelerate! A less formal, but more intuitive interpretation is that we are asking the engine to be able to accelerate at least its own mass.

In the general case, the analogue set of conditions will involve the coupling between all the subsystems. Ultimately, we obtain a set of condition linking the technology (in the form of $\partial m_i / \partial X_k$) to requirement for the mission R_k. It would be surprising if it was any other way.

We now present the proof for lemma 10

Proof. For $\mathcal{F}$ to be a contraction mapping, we must prove the existence of $q \in [0, 1)$ such that

$$\|\mathcal{F}(\vec{Y}) - \mathcal{F}(\vec{Z})\|_a \leq q \cdot \|\vec{Y} - \vec{Z}\|_a \qquad \forall \vec{Y}, \vec{Z} \in \mathbb{R}^{n+}$$

The most important degree of freedom we have is the choice of the norm $\| \cdot \|$. For this proof we will use a modified L_1 norm, based on the c_i coefficients. The only reason for doing so is that it works; there might indeed be better options, but they have escaped us so far. Formally, we define the modified norm as

$$\|\vec{X}\|_c = \sum_{i=1}^{n} c_i \cdot |x_i| \tag{5.36}$$

We begin by working on the left side of the contraction map definition

$$\begin{aligned} &\|\mathcal{F}(\vec{Y}) - \mathcal{F}(\vec{Z})\|_c = \\ =&\|\mathbf{B}_1 \left(\mathbf{A}_{\mathbf{B}_2(\vec{Y})} \vec{r}(t) \right) - \mathbf{B}_1 \left(\mathbf{A}_{\mathbf{B}_2(\vec{Z})}) \vec{r}(t) \right) \|_c = \\ =&\|\max_{t \in \mathcal{T}} \left(\mathbf{A}_{\mathbf{B}_2(\vec{Y})} \cdot \vec{r}(t) \right) - \max_{t \in \mathcal{T}} \left(\mathbf{A}_{\mathbf{B}_2(\vec{Z})} \cdot \vec{r}(t) \right) \|_c \end{aligned} \tag{5.37}$$

However, since $\mathbf{A}_{\mathbf{B}_2(\vec{X})}(\vec{r}(t))$ is the product of a diagonal matrix (which will call $M(\vec{X})$) and the vector $\vec{r}(t)$, we can extract it from the maximum operator(formally $\mathbf{B}_1$). Thus, using the definition of $\vec{R}$, we can eliminate the dependence of time

$$\begin{aligned}
&= \| M(\vec{Y}) \cdot \max_{t\in\mathcal{T}} \, (\vec{r}(t)) - M(\vec{Z}) \cdot \max_{t\in\mathcal{T}} \, (\, \vec{r}(t)) \, \|_c = \\
&= \| M(\vec{Y}) \cdot \vec{R} - M(\vec{Z}) \cdot \vec{R} \|_c = \\
&= \| \left[M(\vec{Y}) - M(\vec{Z}) \right] \cdot \vec{R} \|_c
\end{aligned} \tag{5.38}$$

Consider the structure of M

$$\mathrm{M}(\vec{X}) = \begin{bmatrix} \mathbf{B}_{2,1}(\vec{X}) & 0 & 0 & \dots & 0 \\ 0 & \mathbf{B}_{2,2}(\vec{X}) & 0 & \dots & 0 \\ \dots & \dots & \dots & & \dots \\ 0 & 0 & 0 & \dots & \mathbf{B}_{2,n}(\vec{X}) \end{bmatrix}$$

Then, remembering we have assumed all components of $\mathbf{B}_2$ to be Lipschitz (or to have bounded derivative), we can write

$$\mathrm{M}(\vec{Y}) - \mathrm{M}(\vec{Z}) = \begin{bmatrix} \mathbf{B}_{2,1}(\vec{Y}) - \mathbf{B}_{2,1}(\vec{Z}) & 0 & 0 & \dots & 0 \\ 0 & \mathbf{B}_{2,2}(\vec{Y}) - \mathbf{B}_{2,2}(\vec{Z}) & 0 & \dots & 0 \\ \dots & \dots & \dots & & \dots \\ 0 & 0 & 0 & \dots & \mathbf{B}_{2,n}(\vec{Y}) - \mathbf{B}_{2,n}(\vec{Z}) \end{bmatrix}$$

Since we are interested in *extracting* the difference $Y - Z$ to compare it with the other side, we can overestimate each term in the diagonal of matrix M using the *maximum* slope of increase $\nabla \vec{m}_i$.

$$M(\vec{Y}) - M(\vec{Z}) \leqq \begin{bmatrix} \nabla \vec{m}_1 \cdot (\vec{Y} - \vec{Z}) & 0 & 0 & \dots & 0 \\ 0 & \nabla \vec{m}_2 \cdot (\vec{Y} - \vec{Z}) & 0 & \dots & 0 \\ \dots & \dots & \dots & & \dots \\ 0 & 0 & 0 & \dots & \nabla \vec{m}_n \cdot (\vec{Y} - \vec{Z}) \end{bmatrix}$$

Some care should be taken in the definition of $\nabla \vec{m}_i$. A gross overestimation, and the one we will use, could be

$$\nabla \vec{m}_i \doteq \left(\max \frac{\partial \, \mathbf{B}_{2,i}}{\partial X_1}, \max \frac{\partial \, \mathbf{B}_{2,i}}{\partial X_2}, \dots, \max \frac{\partial \, \mathbf{B}_{2,i}}{\partial X_n} \right)^{\mathsf{T}}$$

To simplify the notation, let us define a difference vector $\vec{X} \doteq \vec{Y} - \vec{Z}$, and as $\mathrm{M}'(\vec{X})$ the above matrix. We can therefore re-write the thesis as

$$\| \mathcal{F}(\vec{Y}) - \mathcal{F}(\vec{Z}) \|_c \leq \| \mathrm{M}'(\vec{X}) \cdot \vec{R} \| \leq q \cdot \| \vec{X} \|_c$$

By applying the $\| \cdot \|_c$ norm based on the vector $\vec{c} \in \mathbb{R}^{n+}$, we have

$$\sum_{i=1}^{n} c_i \cdot |(\nabla \vec{m}_i \cdot \vec{X}) R_i| \leq q \cdot \sum_{i=1}^{n} c_i \cdot |X_i|$$

Note that on the right hand side we have the components of $\vec{X}$, while on the left hand side we have linear functions of $\vec{X}$. To compare the two, we expand the terms on the left. Note that (by hypothesis of non negativity on $\vec{r}(t)$ and non decreasing $\mathbf{B}_2(\vec{X})$), we have $\nabla\vec{m}_i \gg 0$, for each $i = 1, 2, ...n$

$$c_i| \begin{pmatrix} \frac{\partial m_i}{\partial X_1} & \frac{\partial m_i}{\partial X_2} & \dots & \frac{\partial m_i}{\partial X_n} \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{pmatrix} \cdot R_i| \leq c_i \begin{pmatrix} \frac{\partial m_i}{\partial X_1} & \frac{\partial m_i}{\partial X_2} & \dots & \frac{\partial m_i}{\partial X_n} \end{pmatrix} \cdot \begin{pmatrix} |X_1| \\ |X_2| \\ |\dots| \\ |X_n| \end{pmatrix} \cdot R_i$$

Then, since $\vec{X}$ appears only in piece wise absolute value, we can rename it $\vec{X}$ component by component $X_i = |X_i|$. Now we need to prove

$$\sum_{i=1}^{n} c_i \cdot (\nabla\vec{m}_i \cdot \vec{X}) R_i \leq q \cdot \sum_{i=1}^{n} c_i \cdot X_i$$

Expand the components of the difference vector $\vec{X}$

$$\sum_{i=1}^{n} c_i \cdot \left(\sum_{j=1}^{n} \frac{\partial m_i}{\partial X_j} X_j \right) \cdot R_i \leq q \cdot \sum_{i=1}^{n} c_i \cdot X_i \tag{5.39}$$

We regroup according to the components X_i

$$\sum_{i=1}^{n} X_i \cdot \left(\sum_{j=1}^{n} \frac{\partial m_j}{\partial X_i} \cdot c_j R_j \right) - q \cdot \sum_{i=1}^{n} X_i \cdot c_i \leq 0 \quad \Leftrightarrow \quad \sum_{i=1}^{n} X_i \cdot \left(\sum_{j=1}^{n} \frac{\partial m_j}{\partial X_i} \cdot R_j c_j - q \cdot c_i \right) \leq 0 \tag{5.40}$$

Since by definition $X_i \geq 0$, all coefficients in parenthesis must be non positive. Otherwise, a $\vec{X} = \hat{e}_i$ would produce a positive result, and we have to prove that a $\hat{c}$ exists for all $\vec{X} \in \mathbb{R}^n$. Since equation 5.40 must be true for any $\vec{X} \gg= \vec{0}$, it is **necessary and sufficient** to satisfy the n conditions below

$$\begin{cases} \sum_{j=1}^{n} \frac{\partial m_j}{\partial X_1} \cdot R_j c_j - q \cdot c_1 & \leq & 0 \\ \sum_{j=1}^{n} \frac{\partial m_j}{\partial X_2} \cdot R_j c_j - q \cdot c_2 & \leq & 0 \\ \dots & \leq & 0 \\ \sum_{j=1}^{n} \frac{\partial m_j}{\partial X_n} \cdot R_j c_j - q \cdot c_n & \leq & 0 \end{cases} \tag{5.41}$$

Which have the remarkable advantage of not featuring $\vec{X}$. To satisfy the above (Eq. 5.41), we need to prove the existence of a $\hat{c} \in \mathbb{R}^{n++}$ such that:

$$\begin{bmatrix} \frac{\partial m_1}{\partial X_1} \cdot R_1 - q & \frac{\partial m_2}{\partial X_1} \cdot R_2 & \dots & \frac{\partial m_n}{\partial X_1} \cdot R_n \\ \frac{\partial m_1}{\partial X_2} \cdot R_1 & \frac{\partial m_2}{\partial X_2} \cdot R_2 - q & \dots & \frac{\partial m_n}{\partial X_2} \cdot R_n \\ \dots & \dots & \dots & \dots \\ \frac{\partial m_1}{\partial X_n} \cdot R_1 & \dots & \dots & \frac{\partial m_n}{\partial X_n} \cdot R_n - q \end{bmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix} \ll= \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix} \qquad q \in [0, 1) \tag{5.42}$$

□

Condition 5.42 is not yet a simple check; it requires us to prove the existence $\hat{c}$. However, we will provide a direct way to assess if $\vec{c}$ exists in following lemmas.

Remark. The immediate application of Lemma 10 is to assure us that the iterative method is well founded, and will lead the design to the strong optimum design point. However, the result might not be exceedingly interesting in itself due to heavily simplified model used to prove it. On the other hand, more complex models might involve more articulated proofs. To this end, it will be useful to keep in mind that

1. The metric was chosen for convenience; it is not clear whether it is the most general one. Therefore, even if an architecture is not able to meet the hypothesis in Lemma 12, it might still be able to accomplish the mission.

2. Banach fixed point theorem provides a sufficient condition, so if we want to assess for which mission requirements a strong optimum does not exists, we need to use some converse theorems and prove that no complete metric space exists for which $\mathcal{F}$ is a contraction.

If we assume the technology functions $\mathbf{B}_{2,i}$ to be linear in X_i, then the derivatives are constant and the definition of the matrix $\mathbf{Q}$ becomes simpler to obtain. The linear assumption can be justified in two way: either it is exactly true, or it is an approximation.

1. Architectural linearity: cluster typically exhibit linear scaling properties.

2. Linearity as an approximation:
   This could be either interpreted as a linear regression, to capture mean technology laws and then design the system around the strong optimum point, or one could use a linear function as a upper bound for the real function.

5.3.4 Practical methods to prove existence of $\vec{c}$

General methods, typically based on Farkas' lemma, may be used to verify the existence of a suitable (strictly positive) $\vec{c}$. However, we can provide more immediate tests by exploiting the structural propriety of $\mathbf{Q}$. Proofs for the following lemmas are given at the end of the chapter, in section 5.6.

Lemma 11. Given $\mathbf{Q} \in \mathbf{R}^{n \times n}$ with all component off diagonal non negative $\mathbf{Q}_{i,j} \geq 0$, we have that

$$\exists \hat{c} \in \mathbb{R}^{n++} \text{ such that } \mathbf{Q} \cdot \hat{c} \ll 0 \quad \Leftrightarrow \quad q^{(j)}_{j,j} < 0 \quad \forall j = 1, 2, \dots, n \tag{5.43}$$

where the elements $q^{(i)}_{\cdot,\cdot}$ at the ith iteration of a procedure are defined with the recursively as

$$q^{(i)}_{u,p} \doteq q^{(i-1)}_{u,p} \cdot q^{(i-1)}_{i,i} + q^{(i-1)}_{i,p} \cdot q^{(i-1)}_{u,i} \qquad i = 2, 3, \dots n \tag{5.44}$$

in which we are considering the u-th row and p-th column and the initial conditions are that $q^{(1)}_{j,k} = \mathbf{Q}_{j,k}$

Condition 5.43 (above) while being conceptually easy, might be laborious to compute. We thus offer an equivalent algebraic condition.

Lemma 12. Let $\vec{r}(t)$ be mission requirements and $\mathbf{A}_{\vec{m}}, \mathbf{B}_1, \mathbf{B}_2$ as defined in 5.28,5.30 respectively. $\mathcal{F}$ is a contraction mapping if the signs of the minor in a matrix Q alternate as follows

$$sgn\left(\begin{vmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,i} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,i} \\ & \cdots & \cdots & \\ q_{i,1} & q_{i,2} & \cdots & q_{i,i} \end{vmatrix}\right) = -1^i \quad \text{where} \begin{cases} q_{j,j} & \doteq & \max \frac{\partial \mathbf{B}_{2,j}}{\partial \mathrm{X}_j} \cdot \max_{t\in\tau}(r_j(t)) - 1 \\ q_{i,j} & \doteq & \max \frac{\partial \mathbf{B}_{2,j}}{\partial \mathrm{X}_i} \cdot \max_{t\in\tau}(r_j(t)) \end{cases} \tag{5.45}$$

5.4 Consequences of Lemma 10

From this initial result, we can easily expand the model to include more interesting cases. To begin with, we can account for external forces or fluxes which are directly defined by the mission, and are independent of the system characteristics. Subsystem requirements, which previously included only subsystem authority, can be expanded to account for primary and secondary storage, like non rechargeable and rechargeable batteries respectively. This allows us to size also propellant tanks and to model the depletion of on board resources in general. Finally, we can model linear cross couplings and inter dependencies between the subsystems.

5.4.1 External fluxes

We can expand our model to include external disturbances that do not depend on system parameters. For example, knowing the attitude needed for a smallsat during its mission we can estimate the solar heat on its faces. A quantification of this could be expressed as

$$m_s \cdot c_t \cdot \frac{\mathrm{d}}{\mathrm{d}t} T(t) + \underset{\text{mission induced requirement}}{q_{ext}(t)} = \underset{\text{Action of the control system}}{q_{TCS}(t)}$$

We can incorporate a system independent vector $\Phi(t)$ directly into the general equation

$$\mathbf{A}_{\vec{m}}(\vec{r}(t)) \doteq \quad \vec{x}(t) = M_{\vec{m}} \cdot \vec{r}(t) + \Phi(t) = \tag{5.46}$$

It has no effect on the definition on the proof of $\mathcal{F}()$ contraction map as the effect of $\Phi(t)$ cancel out

$$\begin{aligned}
\|\mathcal{F}(\vec{Y}) - \mathcal{F}(\vec{Z})\| &= \|\max_{t\in\mathcal{T}} \left(\mathbf{M}(\vec{Y}) \cdot \vec{r_0}(t) + \vec{\Phi}_0(t)\right) \quad -\max_{t\in\mathcal{T}} \left(\mathbf{M}(\vec{Z}) \cdot \vec{r_0}(t) + \vec{\Phi}_0(t)\right) \| \\
&\leq \|\max_{t\in\mathcal{T}} \left(\mathbf{M}(\vec{Y}) \cdot \vec{r_0}(t) + \vec{\Phi}_0(t) \quad - \mathbf{M}(\vec{Z}) \cdot \vec{r_0}(t) - \vec{\Phi}_0(t)\right) \| \\
&= \|\max_{t\in\mathcal{T}} \left(\mathbf{M}(\vec{Y}) \cdot \vec{r_0}(t) + \cancel{\vec{\Phi}_0(t)} \quad - \mathbf{M}(\vec{Z}) \cdot \vec{r_0}(t) - \cancel{\vec{\Phi}_0(t)}\right) \| \\
&= \|\max_{t\in\mathcal{T}} \left(\mathbf{M}(\vec{Y}) \cdot \vec{r_0}(t) \quad - \mathbf{M}(\vec{Z}) \cdot \vec{r_0}(t)\right) \| \\
&= \|\mathbf{M}(\vec{Y} - \vec{X}) \cdot \vec{R_0}(t)\|
\end{aligned} \tag{5.47}$$

5.4.2 Modeling primary storage

Consider a subsystem which is a simple supplier of some good (current, thrust, torque etc) which comes from an internal storage which can not be resupplied. Then, in order to accomplish the whole mission

$$\int_{t_0}^{t_{\text{end}}} x_i(\tau)\mathrm{d}\tau \leq \text{Cap}_i$$

Which just means that we can not consume more that what we had *designed into the system*; the capacity of the subsystem can not be exceeded.
We want to expand the vector of subsystem requirements to capacity as well, possibly in the

$\max(\cdot)$ form, so that we can build on previous results. To begin with, we can again write each subsystem output as a function of mission requirements $\vec{r}(t)$ and system parameters

$$\mathbf{A}_{\vec{m}}(\vec{r}(t)) = \vec{x}(t) \quad \rightarrow \quad \begin{bmatrix} m_1 & \dots & m_n \end{bmatrix} \cdot \begin{pmatrix} r_1(t) \\ r_2(t) \\ \dots \\ r_n(t) \end{pmatrix} = x_i(t)$$

And, by linearity, we can write

$$\int_{t_0}^{t_{\text{end}}} x_i(\tau)\mathrm{d}\tau = \int_{t_0}^{t_{end}} \sum_{j=1}^{n} m_j \cdot r_j(\tau)\mathrm{d}\tau = \sum_{j=1}^{n} m_j \int_{t_0}^{t_{end}} r_j(\tau)\mathrm{d}\tau$$

Then, given the mission requirements $\vec{r}(t)$, we can include capacity by extending the system model to

$$\begin{bmatrix} \vec{m} & 0 \\ 0 & \vec{m} \end{bmatrix} \cdot \begin{pmatrix} \vec{r}(t) \\ \int_{t_0}^{t_{\mathrm{t}}} \vec{r}(\tau)\mathrm{d}\tau \end{pmatrix} = \begin{pmatrix} \vec{x}(t) \\ \int_0^t \vec{x}(\tau)\mathrm{d}\tau \end{pmatrix} \tag{5.48}$$

Hence the problem can be formulated under the same shape. Note that, since every component $x_i(t)$ is non negative, we have that

$$\max_{t \in \mathcal{T}} \int_0^t \vec{x}(\tau)\mathrm{d}\tau = \int_0^{t_{end}} \vec{x}(\tau)\mathrm{d}\tau$$

Which enforces the idea that no re-supply is allowed.

5.4.3 Secondary capacity

We want to consider secondary storage system, where resources may be reintegrated during the mission. This abstraction models the fact that the solar panel may recharge the battery or that in some circumstances, refueling might be allowed. Requirements on secondary capacity will clearly be a function of the net effect of output and input signals over the mission duration

$$Cap = f\left(\int_0^t \text{output }(\tau) - \text{input}(\tau)\mathrm{d}\tau\right) \tag{5.49}$$

To generalize for the whole system, we add to Eq. (5.48) a vector of *influx* $\vec{\Phi} \in \mathbb{R}^{+n}$,

$$\begin{bmatrix} \mathbf{S}(\vec{X}) & 0 \\ 0 & \mathbf{S}(\vec{X}) \end{bmatrix} \cdot \begin{pmatrix} \vec{r}(t) \\ \int_0^t \vec{r}(\tau)\mathrm{d}\tau \end{pmatrix} - \begin{pmatrix} 0 \\ \vec{\Phi}(t) \end{pmatrix} = \begin{pmatrix} \vec{x}(t) \\ \int_0^t \vec{x}(\tau)\mathrm{d}\tau \end{pmatrix}$$

In which, as we have already seen for primary capacity

$$\int_0^t \mathrm{s}_i(\tau)\mathrm{d}\tau = \sum_{j=1} S_{i,j} \int_0^t r_j(\tau)\mathrm{d}\tau \quad \text{and} \quad \Phi_i(t) \doteq \int_0^t \mathrm{p}_i(\tau)\mathrm{d}\tau$$

So far we have just added a vector to the familiar representation. However, we have to be careful in imposing the constraints on capacity; it would be a mistake to impose

$$Cap = \max_{t \in \mathcal{T}} \left(\int_0^t \text{output}(\tau) - \text{input}(\tau)\mathrm{d}\tau\right)$$

Consider the case in which the inputs are, on average, bigger than the output. The integral in Eq. (5.49) would be mostly negative, with perhaps some small positive value, which we would identify as the needed capacity. We would expect this to happen when the solar panel provides a lot of power, which can be used directly by the system. However, by setting the capacity using a simple max on the integral, we are neglecting the effect of saturation; while the battery can not store infinite energy, the signal can become arbitrarily negative without any effect on the request on battery capacity. An example of this behavior is shown in Fig 5.6, where using max to set the requirement for capacity underestimates the real value by half.

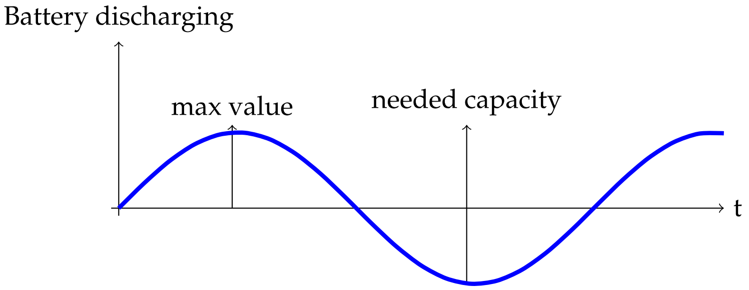


Figure 5.6: Net energy flow; positive sign indicates energy leaving the battery

The real condition we need to set on the capacity is not the maximum, but rather *maximum excursion*, defined as

$$\max_{t\in T}\Delta f(t) \doteq \max_{t\in T} f(t) - \min_{t\in T} f(t)$$

Then, we need to adjust the definition of $\mathcal{F}(\cdot)$ as follows

$$\mathcal{F}(\vec{X}) \doteq \begin{pmatrix} \max\limits_{t\in\mathcal{T}} & \mathbf{S}(\vec{X})\cdot\vec{r}(t) \\ \max\limits_{t\in\mathcal{T}}\Delta & \mathbf{S}(\vec{X})\cdot\int_0^t \vec{r}(\tau)\mathrm{d}\tau - \vec{\Phi}(t) \end{pmatrix}$$

Is $\mathcal{F}(\cdot)$ still a contraction map under the same assumptions? The answer is positive.

Proof. To prove this, notice that, if $f(t) \in \mathbb{R}^{+n}$ for all $t \in \mathcal{T}$ and there exists a t_0 such that $f(t_0) = \vec{0}$, then

$$\max_{t\in\mathcal{T}}\Delta f(t) = \max_{t\in\mathcal{T}} f(t) - \min_{t\in\mathcal{T}} f(t) = \max_{t\in\mathcal{T}} f(t) - f(t_0) = \max_{t\in\mathcal{T}} f(t)$$

Therefore, if we assume that $\vec{r}(t=0) = \vec{0}$, we can obtain a more homogeneous definition of $\mathcal{F}$

$$\mathcal{F}(\vec{X}) \doteq \max_{t\in\mathcal{T}}\Delta \begin{pmatrix} \mathbf{S}(\vec{X})\cdot\vec{r}(t) \\ \mathbf{S}(\vec{X})\cdot\int_0^t \vec{r}(\tau)\mathrm{d}\tau \quad -\vec{\Phi}(t) \end{pmatrix} = \max_{t\in\mathcal{T}}\Delta \left(\mathbf{S_0}(\vec{X})\cdot\vec{r}_0(t) - \vec{\Phi}_0(t)\right)$$

In which we have renamed some matrices to improve readability. Now, we want to prove that

$$\|\mathcal{F}(\vec{Y}) - \mathcal{F}(\vec{Z})\| \leq q\|\vec{Y} - \vec{Z}\| \quad q \in [0,1)$$

Notice that:

$$\begin{aligned}
\|\mathcal{F}(\vec{Y}) - \mathcal{F}(\vec{Z})\| &= \|\max_{t\in\mathcal{T}} \Delta \left(\mathbf{S_0}(\vec{Y}) \cdot \vec{r}_0(t) - \vec{\Phi}_0(t)\right) \quad - \max_{t\in\mathcal{T}} \Delta \left(\mathbf{S_0}(\vec{Z}) \cdot \vec{r}_0(t) - \vec{\Phi}_0(t)\right) \| \\
&\leq \|\max_{t\in\mathcal{T}} \Delta \left(\mathbf{S_0}(\vec{Y}) \cdot \vec{r}_0(t) - \vec{\Phi}_0(t) \quad - \mathbf{S_0}(\vec{Z}) \cdot \vec{r}_0(t) + \vec{\Phi}_0(t)\right) \| \\
&= \|\max_{t\in\mathcal{T}} \Delta \left(\mathbf{S_0}(\vec{Y}) \cdot \vec{r}_0(t) - \cancel{\vec{\Phi}_0(t)} \quad - \mathbf{S_0}(\vec{Z}) \cdot \vec{r}_0(t) + \cancel{\vec{\Phi}_0(t)}\right) \| \\
&= \|\max_{t\in\mathcal{T}} \left(\mathbf{S_0}(\vec{Y}) \cdot \vec{r}_0(t) \quad - \mathbf{S_0}(\vec{Z}) \cdot \vec{r}_0(t)\right) \| \\
&= \|\mathbf{S_0}(\vec{Y} - \vec{X}) \cdot \vec{R}_0(t)\|
\end{aligned}$$

From which we can simply proceed as in the first lemma. □

The proof above might be complemented with a simple exercise to prove that

$$\max_{x\in X} \Delta \{f(x)\} - \max_{x\in X} \Delta \{g(x)\} \leq \max_{x\in X} \Delta \{f(x) - g(x)\}$$

$$\begin{aligned}
\max_{x\in X} \Delta \{f(x) + g(x)\} &= \max_{x\in X} \{f(x) + g(x)\} - \min_{x\in X} \{f(x) + g(x)\} \\
&\leq \max_{x\in X} \{f(x)\} + \max_{x\in X} \{g(x)\} - \min_{x\in X} \{f(x) + g(x)\} \\
&\leq \max_{x\in X} \{f(x)\} + \max_{x\in X} \{g(x)\} - \min_{x\in X} \left\{f(x) + \min_{x\in X} g(x)\right\} \\
&\leq \max_{x\in X} \{f(x)\} - \min_{x\in X} \{f(x)\} + \max_{x\in X} \{g(x)\} - \min_{x\in X} \{\ g(x)\} \\
&\leq \max_{x\in X} \Delta \{f(x)\} + \max_{x\in X} \Delta \{g(x)\}
\end{aligned}$$

Therefore, as in the case above

$$\max_{x\in X} \Delta \{f(x)\} - \max_{x\in X} \Delta \{g(x)\} \leq \max_{x\in X} \Delta \{f(x) - g(x)\}$$

5.5 Non ideal systems central Lemma

We now want to account for the inter dependencies between subsystems. An engine will produce torque as a main output but also excess heat as a by product, the temperature control system will consume current produced by the power system etc.

All of these interactions contribute negatively to the design of the system because they produce higher requirements for each subsystem, which therefore needs more authority/capacity etc. This is both a cautionary assumption (as if they contribute positively, all the better) but also a formal hypothesis which will be essential for this formulation of the algorithm. A generic *non ideal* system might respond to mission requirements as

$$\begin{pmatrix} F(t) \\ i(t) \\ q(t) \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\eta_1 & 1 & -\eta_2 \\ -\eta_3 & -\eta_4 & 1 \end{bmatrix} \begin{pmatrix} F_e(t) \\ i_{ps}(t) \\ q_{ts}(t) \end{pmatrix} \tag{5.50}$$

To understand Eq (5.50), let us consider the first row. The force that is required from the system must be matched by the force produced by the engine. On the second row, the current requested by

the payload (for example) must be produced by the power system. However, both the engine and the thermal control system require current themselves, so a surplus of current is needed according to the requests on the other subsystems. In fact, if more force is requested, more current will be drawn, regardless of the payload $i(t)$ requirement.
Finally, consider the third row. Assume that the payload requires some heat to be removed. Then, as before, the thermal control system will have to extract that heat, plus the heat from the engine and the power system. This again assumes anti symmetric operations for each subsystem. Since the parasitic components are expressed by the coefficients η_i, if we were able to set $\eta_i = 0$ we would obtain the ideal system, which is the one with the lowest requirements.

As in the previous section, we want to express mission requirements independently of the system static characteristics

$$\begin{pmatrix} F(t) \\ i(t) \\ q(t) \end{pmatrix} = \begin{pmatrix} m_s & \cdot & a(t) \\ 1 & \cdot & i(t) \\ m_s c_t & \cdot & q(t) \end{pmatrix} = \begin{bmatrix} m_s & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & m_s \cdot c_t \end{bmatrix} \cdot \begin{pmatrix} a(t) \\ i(t) \\ q(t) \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\eta_1 & 1 & -\eta_2 \\ -\eta_3 & -\eta_4 & 1 \end{bmatrix} \begin{pmatrix} F_e(t) \\ i_{ps}(t) \\ q_{ts}(t) \end{pmatrix}$$

It will be useful to decompose the system using the two auxiliary matrices, as

$$[\mathbf{M}]_{\vec{X}}\vec{r}(t) = [\eta]\vec{x}(t) \tag{5.51}$$

At this point, we allow $\mathbf{M}$ to be a function of $\vec{X}$, the system parameters, while the matrix $[\eta]$ depends on the chosen technology, so it is governed by independent variables. The fundamental assumption for this model is that η allows for an inverse and that at least one of such inverses has only non negative components. Note that the choice of η^{-1} is implied in the operation of the system; we need to know which output to ask each subsystem to obtain a certain system output. At the beginning of the chapter we have used this condition as the characterization of the mission worthy set $\mathcal{A}$. Moreover, if η^{-1} had some negative component, it would mean that, for some $\vec{r}$, it would be necessary to operate the actuator in reverse (as in $x_i < 0$); this is clearly unacceptable in the engineering model.

The main result is the following

Lemma 13. Assuming that η^{-1} is non negative and the hypothesis of the previous model, the system

$$\mathbf{M}(X) \cdot \vec{r}(t) = \eta\vec{x}(t) \quad \mathcal{F}(X) \doteq \max_{t\in\mathcal{T}} \mathbf{S}(\vec{X}) \cdot \vec{r}(t) = \max_{t\in\mathcal{T}} \eta^{-1} \cdot M(\vec{X}) \cdot \vec{r}(t) \tag{5.52}$$

has a strong optimum point if there exists a vector $\vec{c} \in \mathbb{R}^n$ such that

$$\eta^{-1}\mathbf{S}(X) \cdot \vec{c} \ll 0 \tag{5.53}$$

In the proof we will use some lemmas, the proof of which is presented in section 5.6.1.

The proof is very similar to the one we have already presented, although a bit more articulated due to the fact that the matrix $\mathbf{S}(X)$ is no longer diagonal.

Proof. Once again, we need to prove that the new $\mathcal{F}$ is a contraction mapping so that there exist $q \in [0, 1)$ such that

$$\|\mathcal{F}(\vec{Y}) - \mathcal{F}(\vec{Z})\| \leq q \cdot \|\vec{Y} - \vec{Z}\|$$

$$
\begin{array}{rcccc}
\|\mathcal{F}(\vec{Y})-\mathcal{F}(\vec{Z})\| & = & \|\max\limits_{t\in\mathcal{T}}\left(S(\vec{Y})\cdot\vec{r}(t)\right) & - & \max\limits_{t\in\mathcal{T}}\left(S(\vec{Z})\cdot\vec{r}(t)\right)\| \\
 & \leq & \|\max\limits_{t\in\mathcal{T}}\left(S(\vec{Y})\cdot\vec{r}(t)\right. & - & \left.S(\vec{Z})\cdot\vec{r}(t)\right)\| \\
 & = & \|\max\limits_{t\in\mathcal{T}}\left(\eta^{-1}\cdot M(\vec{Y})\cdot\vec{r}(t)\right. & - & \left.\eta^{-1}\cdot M(\vec{Z})\cdot\vec{r}(t)\right)\| \\
 & \leq & \|\eta^{-1}\cdot\left(M(\vec{Y})-M(\vec{Z})\right) & \cdot & \max\limits_{t\in\mathcal{T}}\left(\vec{r}(t)\right)\| \\
 & = & \|\eta^{-1}\cdot M(\vec{Y}-\vec{Z})\cdot\vec{r}\| & &
\end{array}
$$

We want to prove that, for some norm $\|\cdot\|$

$$
\|\eta^{-1}M(\vec{Y}-\vec{Z})\cdot\vec{R}\|\leq q\|\vec{Y}-\vec{Z}\|
$$

Again, let us chose the norm induced by the vector $\vec{c}$,

$$
\sum_{i=1}^{n}c_i\cdot|\sum_{j=1}^{n}\gamma_{i,j}\cdot M_{j,j}(\vec{Y}-\vec{Z})\cdot R_j|\leq q\cdot\sum_{i=1}^{n}c_i\cdot|Y_i-Z_i| \tag{5.54}
$$

Due to the non negativity of R_i and $\gamma_{i,j}$ (where $\left\{\gamma_{i,j}\right\}=\eta^{-1}$) we can write

$$
\sum_{i=1}^{n}c_i\cdot|\sum_{j=1}^{n}\gamma_{i,j}\cdot M_{j,j}(\vec{Y}-\vec{Z})\cdot R_j|\leq\sum_{i=1}^{n}c_i\cdot\sum_{j=1}^{n}\gamma_{i,j}\cdot|M_{j,j}(\vec{Y}-\vec{Z})|\cdot R_j
$$

Also, remember that

$$
M(\vec{Y})-M(\vec{Z})=\begin{bmatrix}\nabla\vec{m}_1\cdot(\vec{Y}-\vec{Z}) & 0 & \dots & 0\\ 0 & \nabla\vec{m}_2\cdot(\vec{Y}-\vec{Z}) & \dots & 0\\ 0 & 0 & \dots & \nabla\vec{m}_n\cdot(\vec{Y}-\vec{Z})\end{bmatrix}
$$

Therefore

$$
|M_{j,j}(\vec{Y}-\vec{Z})|=|\sum_{k=1}^{n}\frac{\partial m_j}{\partial X_k}\cdot(Y_k-Z_k)|\leq\sum_{k=1}^{n}\frac{\partial m_j}{\partial X_k}\cdot X_k \qquad X_k\doteq|Y_k-Z_k|
$$

We can prove Eq. (5.54) by proving that

$$
\sum_{i=1}^{n}c_i\cdot\sum_{j=1}^{n}\gamma_{i,j}\cdot\sum_{k=1}^{n}\frac{\partial m_j}{\partial X_k}\cdot X_k\cdot R_j\leq q\cdot\sum_{i=1}^{n}c_i\cdot X_i
$$

We group it, as in the previous case, by X_i

$$
\sum_{k=1}^{n}X_k\cdot\left(\sum_{i=1}^{n}c_i\cdot\left(\sum_{j=1}^{n}\gamma_{i,j}\frac{\partial m_j}{\partial X_k}\cdot R_j\right)\right)-q\cdot\sum_{k=1}^{n}c_k\cdot X_k\leq 0 \tag{5.55}
$$

$$
\sum_{k=1}^{n}X_k\cdot\left(\sum_{i=1}^{n}c_i\cdot\left(\sum_{j=1}^{n}\gamma_{i,j}\frac{\partial m_j}{\partial X_k}\cdot R_j\right)-q\cdot c_k\right)\leq 0 \tag{5.56}
$$

Which again is independent of X_k, as they are all positive, and can be solved only if there exists a vector $\vec{c} \gg \vec{0}$ such that

$$\begin{cases} \sum_{i=1}^{n} c_i \cdot \left(\sum_{j=1}^{n} \gamma_{i,j} \frac{\partial m_j}{\partial X_1} \cdot R_j \right) - q \cdot c_1 & \leq & 0 \\ \sum_{i=1}^{n} c_i \cdot \left(\sum_{j=1}^{n} \gamma_{i,j} \frac{\partial m_j}{\partial X_2} \cdot R_j \right) - q \cdot c_2 & \leq & 0 \\ \dots & \leq & 0 \\ \sum_{i=1}^{n} c_i \cdot \left(\sum_{j=1}^{n} \gamma_{i,j} \frac{\partial m_j}{\partial X_n} \cdot R_j \right) - q \cdot c_n & \leq & 0 \end{cases} \tag{5.57}$$

Eq. 5.57 is equivalent to prove existence of $\hat{c} \in \mathbb{R}^{n++}$ such that $\mathbf{Q} \cdot \vec{c} \ll= \vec{0}$, where

$$\mathbf{Q} \doteq \begin{bmatrix} \sum_{j=1}^{n} \gamma_{1,j} \frac{\partial m_j}{\partial X_1} \cdot R_j - q & \sum_{j=1}^{n} \gamma_{2,j} \frac{\partial m_j}{\partial X_1} \cdot R_j & \dots & \sum_{j=1}^{n} \gamma_{n,j} \frac{\partial m_j}{\partial X_1} \cdot R_j \\ \sum_{j=1}^{n} \gamma_{1,j} \frac{\partial m_j}{\partial X_2} \cdot R_j & \sum_{j=1}^{n} \gamma_{2,j} \frac{\partial m_j}{\partial X_2} \cdot R_j - q & \dots & \sum_{j=1}^{n} \gamma_{n,j} \frac{\partial m_j}{\partial X_2} \cdot R_j \\ \dots & \dots & \dots & \dots \\ \sum_{j=1}^{n} \gamma_{1,j} \frac{\partial m_j}{\partial X_n} \cdot R_j & \dots & \dots & \sum_{j=1}^{n} \gamma_{n,j} \frac{\partial m_j}{\partial X_n} \cdot R_j - q \end{bmatrix} \tag{5.58}$$

□

5.6 Proofs

Lemma 11

Given $\mathbf{Q} \in \mathbf{R}^{n\times n}$ with all components off diagonal non negative $\mathbf{Q}_{k,j} \geq 0 \; k \neq j$, we have that

$$\exists \hat{c} \in \mathbb{R}^{n++} \text{ such that } \mathbf{Q} \cdot \hat{c} \ll 0 \quad \Leftrightarrow \quad q^{(j)}_{j,j} < 0 \quad \forall j = 2, \ldots, n \tag{5.59}$$

Where the elements in the ith iteration of the $\mathbf{Q}$ matrix can be expressed by the recursive formula

$$q^{(i)}_{u,p} \doteq q^{(i-1)}_{u,p} \cdot q^{(i-1)}_{i,i} + q^{(i-1)}_{i,p} \cdot q^{(i-1)}_{u,i} \qquad i = 2, 3, \ldots n \tag{5.60}$$

in which we are considering the u-th row and p-th column and the initial conditions are that $q^{(1)}_{j,k} = \mathbf{Q}_{j,k}$

Proof. For ease of notation, we will express $q_{i,j}$ in absolute value and use minus signs to mark the negative components. We write Q as

$$\text{Q} \doteq \begin{bmatrix} -q_{1,1} & q_{1,2} & q_{1,3} & \cdots & q_{1,n} \\ q_{2,1} & -q_{2,2} & q_{2,3} & \cdots & q_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ q_{n,1} & q_{n,2} & q_{n,3} & \cdots & -q_{n,n} \end{bmatrix} \qquad q_{j,k} > 0 \, \forall j, k$$

We notice that, if Q was diagonal, $\hat{c}$ could exists if and only if all elements on the diagonal were strictly negative

$$\begin{bmatrix} -q_{1,1} & 0 & \ldots & 0 \\ 0 & -q_{2,2} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & -q_{n,n} \end{bmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \ldots \\ c_n \end{pmatrix} \ll \begin{pmatrix} 0 \\ 0 \\ \ldots \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \forall\, i = 1, \ldots, n \quad (-q_{i,i}) \cdot c_i < 0 \tag{5.61}$$

Using Gauss elimination, we can produce a diagonal problem $Q^\star \cdot \vec{c} \ll 0$ which is equivalent to the original if the n conditions 5.59 on the sign of the diagonal elements of $Q^\star$ are satisfied.

Assume that there exists a $\hat{c}$ able to satisfy equation 5.59. Let $\vec{\varepsilon}$ be its image through Q, $\vec{\varepsilon} \doteq \text{Q} \cdot \hat{c}$. By hypothesis each component of $\vec{\varepsilon}$ is strictly negative. As for the elements of Q, the components of $\vec{\varepsilon}$ will be expressed as their module and a minus sign. Since we are operating on the system $\text{Q}|\vec{\varepsilon}$, we can omit the vector $\hat{c}$. The first step is to pre-multiply the second row by $q_{1,1}$ (which is strictly positive) and add the first one pre-multiplied[5] by $q_{2,1}$. Hence $\text{II}' = q_{1,1} \cdot \text{II} + q_{2,1} \cdot \text{I}$

$q_{1,1} \cdot \text{II} \quad +$	$q_{2,1}q_{1,1} \quad +$	$-q_{2,2}q_{1,1} \quad +$	$\ldots \quad +$	$q_{2,n}q_{1,1} \quad +$	$-\varepsilon_2 \cdot q_{1,1} \quad +$
$q_{2,1} \cdot \text{I} \quad =$	$-q_{2,1}q_{1,1} \quad =$	$q_{2,1}q_{1,2} \quad =$	$\ldots \quad =$	$q_{2,1}q_{1,n} \quad =$	$-\varepsilon_1 \cdot q_{2,1} \quad =$
II'	0	$-q_{2,2}q_{11} + q_{1,2}q_{2,1}$	$\ldots$	$q_{2,n}q_{11} + q_{1,n}q_{2,1}$	$-\varepsilon_2 \cdot q_{11} - \varepsilon_1 \cdot q_{2,1}$

Checking the signs, on the lhs, we have added only positive components, except for the first entry $-q_{1,1}q_{2,1}$, which is now zero. All $\text{Q}'_{2,j}$ with $j \neq 2$ were already positive so they remain

[5] Notice that, if $q_{2,1} = 0$ the operation is pointless, but legal.

positive. On the rhs, we added two negative quantities, so the sign does not change either. Then, it must be true that

$$Q'_{2,2} = \quad -q_{2,2}q_{11} + q_{1,2}q_{2,1} < 0 \tag{5.62}$$

Otherwise, by applying $\hat{c}$, we would have the scalar product of two positive vectors ($\hat{c}$ and $Q'_{2,i} = (Q'_{2,1}, Q'_{2,2}, \ldots, Q'_{2,n})$) resulting in a negative value ($-\varepsilon_2 \cdot q_{1,1} - \varepsilon_1 \cdot q_{2,1}$), which is impossible[6]. We have established that, if $\hat{c}$ exists, it must be true that $Q'_{2,2} < 0$. After this first operation we have

$$\left[\begin{array}{ccccc|c} -q_{1,1} & q_{1,2} & q_{1,3} & \cdots & q_{1,n} & -\varepsilon_1 \\ 0 & -q_{2,2}q_{11} + q_{1,2}q_{2,1} & q_{2,3}q_{1,1} + q_{1,3}q_{2,1} & \cdots & q_{2,n}q_{1,1} + q_{1,n}q_{2,1} & -\varepsilon_2 \cdot q_{1,1} - \varepsilon_1 \cdot q_{2,1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ q_{n,1} & q_{n,2} & q_{n,3} & \cdots & -q_{n,n} & -\varepsilon_n \end{array}\right]$$

We want to iterate the process. First, we can perform the operation to clear the first component of every row

$$\begin{bmatrix} -q_{1,1} & q_{1,2} & q_{1,3} & \cdots & q_{1,n} \\ 0 & q'_{2,2} & q'_{2,3} & \cdots & q'_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & q'_{n,2} & q'_{n,3} & \cdots & q'_{n,n} \end{bmatrix} = \begin{bmatrix} -\varepsilon_1 \\ -\varepsilon_2 \cdot q_{11} - \varepsilon_1 \cdot q_{2,1} \\ \cdots \\ -\varepsilon_n \cdot q_{1,1} - \varepsilon_1 q_{n,1} \end{bmatrix} \qquad q'_{u,p} \doteq q_{u,p} \cdot q_{1,1} + q_{1,p} \cdot q_{u,1}$$

For every row, we have to impose the condition of negativity on the diagonal element, which generalizes 5.62,

$$Q'_{j,j} = \quad -q_{j,j}q_{1,1} + q_{1,j}q_{j,1} < 0 \quad \forall j = 2, \ldots n$$

To obtain the upper triangular form of Q, we iterate the process on every row. We need to guarantee that the elements on the diagonal of the final matrix remain strictly negative. On the rhs, we will have a sum of negative components, which therefore remains negative.

$$Q^{(n-1)} = \begin{bmatrix} -q_{1,1} & q_{1,2} & q_{1,3} & \cdots & q_{1,n} \\ 0 & q'_{2,2} & q'_{2,3} & \cdots & q'_{2,n} \\ 0 & 0 & q''_{3,3} & \cdots & q''_{3,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & q^{(n)}_{n,n} \end{bmatrix} \qquad \vec{\varepsilon}^{\,(n-1)} = \begin{bmatrix} -\varepsilon_1 \\ -\varepsilon_2 \cdot q_{1,1} - \varepsilon_1 \cdot q_{2,1} \\ -\sum_{j=1}^{3} \varepsilon_j \alpha_j \\ \cdots \\ -\sum_{j=1}^{n} \varepsilon_j \cdot \alpha_j \end{bmatrix}$$

The elements in the ith iteration of the Q matrix can be expressed by the recursive formula

$$Q^{(i)}_{u,p} = \quad q^{(i)}_{u,p} \doteq q^{(i-1)}_{u,p} \cdot q^{(i-1)}_{i,i} + q^{(i-1)}_{i,p} \cdot q^{(i-1)}_{u,i}$$

in which we are considering the u-th row and p-th column.

Going from the upper triangular $Q^{(n-1)}$ to the diagonal form of Q is immediate, and does not change the elements on the diagonal. Starting from the last row, element $q^{(n)}_{n,n}$ can be added to the elements above and clear the whole column. On the ε side, we are adding negative quantities

[6] This is a linear system, so rows addition in the system $Q \cdot c = \varepsilon$ maintain the equality and multiplication of a row by a positive scalar maintains the sign

to negative quantities, so the sign does not change. We then repeat the procedure with the rows above. In the resulting diagonal system, the right hand side is negative,

$$\begin{bmatrix} -q_{1,1} & 0 & 0 & \dots & 0 \\ 0 & q'_{2,2} & 0 & \dots & 0 \\ 0 & 0 & q''_{3,3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & q^{(n)}_{n,n} \end{bmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \dots \\ c_n \end{pmatrix} = \begin{bmatrix} -\sigma_1 \\ -\sigma_2 \\ -\sigma_3 \\ \dots \\ -\sigma_n \end{bmatrix}$$

Where we know the signs of the coefficients, therefore we can say that $\vec{c}$ exists positive and finally, we know that the same $\vec{c}$ is also a solution for the initial system.

Therefore, we have an equivalent condition to 5.42

$$\exists \hat{c} \in \mathbb{R}^{n++} \text{ such that } \mathrm{Q}\hat{c} \ll 0 \quad \Leftrightarrow \quad q^{(j)}_{j,j} < 0 \quad \forall j = 2, \dots, n \tag{5.63}$$

□

Lemma 12

Condition 5.43 (above) while being conceptually easy, it is a bit complex to compute. The final step is to refine it into a more algebraic condition. The proof of Lemma 12 follows.

We know that, if we can reduce the Q matrix to an upper triangular form with negative elements on the diagonal, the suitable vector $\hat{c} \gg 0$ exists, and we also know a method to obtain the upper triangular matrix from $\mathrm{Q}^{(n)}$.

$$\mathrm{Q} = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ q_{n,1} & q_{n,2} & \cdots & q_{n,n} \end{pmatrix} \quad \sim \quad \mathrm{Q}^{(n)} \doteq \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\ 0 & q^{(2)}_{2,2} & \cdots & q^{(2)}_{2,n} \\ 0 & 0 & \dots & \dots \\ 0 & 0 & \dots & q^{(n)}_{n,n} \end{pmatrix}$$

Note that in this case, the convention on the signs of $q_{u,k}$ is different than in the previous case! Now they are not all positive, they are simply the element of the matrix. With this notation the recursive formula to compute the elements is

$$q^{(1)}_{u,k} \doteq q_{u,k} \qquad q^{(i)}_{u,k} \doteq -q^{(i-1)}_{u,k} \cdot q^{(i-1)}_{i-1,i-1} + q^{(i-1)}_{i-1,k} \cdot q^{(i-1)}_{u,i-1} \tag{5.64}$$

So we need to ask that $q^{(i)}_{i,i} < 0$ for all $i = 1, ..., n$. In order to have a more clear representation of such conditions, let us write the analytical expressions up to $n = 3$.

$$\begin{array}{lll} i = 1 & q_{1,1} < 0 & \\ i = 2 & q^{(2)}_{2,2} < 0 & \Rightarrow \quad -q^{(1)}_{2,2} \cdot q^{(1)}_{1,1} + q^{(1)}_{1,2} \cdot q^{(1)}_{2,1} < 0 \\ i = 3 & q^{(3)}_{3,3} < 0 & \Rightarrow \quad -q^{(2)}_{3,3} \cdot q^{(2)}_{2,2} + q^{(2)}_{2,3} \cdot q^{(2)}_{3,2} < 0 \end{array}$$

where, using the recursive formula 5.64

$$\begin{array}{rlrl} q^{(2)}_{2,2} = & -q_{2,2}q_{1,1} + q_{1,2}q_{2,1} & \qquad q^{(2)}_{2,3} = & -q_{2,3}q_{1,1} + q_{1,3}q_{2,1} \\ q^{(2)}_{3,2} = & -q_{3,2}q_{1,1} + q_{1,2}q_{3,1} & \qquad q^{(2)}_{3,3} = & -q_{3,3}q_{1,1} + q_{1,3}q_{3,1} \end{array}$$

$q_{3,3}^{(3)} = -(-q_{3,3}q_{1,1} + q_{1,3}q_{3,1}) \cdot (-q_{2,2}q_{1,1} + q_{1,2}q_{2,1}) + (-q_{2,3}q_{1,1} + q_{1,3}q_{2,1}) \cdot (-q_{3,2}q_{1,1} + q_{1,2}q_{3,1})$
Which, after tedious algebraic manipulations, can be written as

$$q_{3,3}^{(3)} = -q_{1,1}[q_{1,1}(q_{2,2}q_{3,3} - q_{2,3}q_{3,2}) - q_{1,2}(q_{3,3}q_{2,1} - q_{2,3}q_{3,1}) + q_{1,3}(q_{2,1}q_{3,2} - q_{3,1}q_{2,2})]$$

$$q_{3,3}^{(3)} = -q_{1,1} \cdot \begin{vmatrix} q_{1,1} & q_{1,2} & q_{1,3} \\ q_{2,1} & q_{2,2} & q_{2,3} \\ q_{3,1} & q_{3,2} & q_{3,3} \end{vmatrix} \quad \Leftrightarrow \quad sgn(q_{3,3}^{(3)}) = sgn\left(\begin{vmatrix} q_{1,1} & q_{1,2} & q_{1,3} \\ q_{2,1} & q_{2,2} & q_{2,3} \\ q_{3,1} & q_{3,2} & q_{3,3} \end{vmatrix}\right)$$

Since $q_{1,1}$ needs to be negative as well. We can express the sign of $q_{3,3}^{(3)}$ as a function of the alternating sign of the j-th minor in the initial matrix.

$$sgn(q_{3,3}^{(3)}) \quad = \quad -sgn\left(\begin{vmatrix} q_{2,2}^{(2)} & q_{2,3}^{(2)} \\ q_{3,2}^{(2)} & q_{3,3}^{(2)} \end{vmatrix}\right) \quad = \quad sng\left(\begin{vmatrix} q_{1,1} & q_{1,2} & q_{1,3} \\ q_{2,1} & q_{2,2} & q_{2,3} \\ q_{3,1} & q_{3,2} & q_{3,3} \end{vmatrix}\right)$$

We wish to prove that the sign of $q_{j,j}^{(j)}$ is given by the sign of the proper determinant in the initial Q matrix,

$$q_{i,i}^{(i)} = (-1)^{i+1} \cdot sgn\left(\begin{vmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,i} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,i} \\ & \cdots & \cdots & \\ q_{i,1} & q_{i,2} & \cdots & q_{i,i} \end{vmatrix}\right) \qquad i = 1, \ldots, n \tag{5.65}$$

We prove it by induction. The basis for the induction is given by the definition of $q_{i,i}^{(i)}$

$$q_{i,i}^{(i)} = -\left(\begin{vmatrix} q_{i-1,i-1}^{(i-1)} & q_{i-1,i}^{(i-1)} \\ q_{i,i-1}^{(i-1)} & q_{i,i}^{(i-1)} \end{vmatrix}\right)$$

The inductive step is to prove that

$$sgn\left(\begin{vmatrix} q_{i-k,i-k}^{(i-k)} & \cdots & q_{i-k,i}^{(i-k)} \\ \cdots & \cdots & \cdots \\ q_{i,i-k}^{(i-k)} & \cdots & q_{i,i}^{(i-k)} \end{vmatrix}\right) = -sgn\left(\begin{vmatrix} q_{i-k-1,i-k-1}^{(i-k-1)} & \cdots & \cdots & q_{i-k-1,i}^{(i-k-1)} \\ q_{i-k,i-k-1}^{(i-k-1)} & q_{i-k,i-k}^{(i-k-1)} & \cdots & q_{i-k,i}^{(i-k-1)} \\ \cdots & \cdots & \cdots & \\ q_{i,i-k-1}^{(i-k-1)} & q_{i,i-k}^{(i-k-1)} & \cdots & q_{i,i}^{(i-k-1)} \end{vmatrix}\right)$$

By imposing an equivalence in value, we are surely maintaining an equivalence of sign. Then the following equivalences (where $\star$ denotes any number) between determinants are easily verified.

$$\begin{vmatrix} q_{i-k,i-k}^{(i-k)} & \cdots & q_{i-k,i}^{(i-k)} \\ \cdots & \cdots & \cdots \\ q_{i,i-k}^{(i-k)} & \cdots & q_{i,i}^{(i-k)} \end{vmatrix} = \begin{vmatrix} 1 & \star & \cdots & \star \\ 0 & q_{i-k,i-k}^{(i-k)} & \cdots & q_{i-k,i}^{(i-k)} \\ 0 & \cdots & \cdots & \cdots \\ 0 & q_{i,i-k}^{(i-k)} & \cdots & q_{i,i}^{(i-k)} \end{vmatrix} = \frac{1}{q_{i-k-1,i-k-1}^{(i-k-1)}} \begin{vmatrix} q_{i-k-1,i-k-1}^{(i-k-1)} & \star & \cdots & \star \\ 0 & q_{i-k,i-k}^{(i-k)} & \cdots & q_{i-k,i}^{(i-k)} \\ 0 & \cdots & \cdots & \cdots \\ 0 & q_{i,i-k}^{(i-k)} & \cdots & q_{i,i}^{(i-k)} \end{vmatrix}$$

We have expanded the minor at the step $(i-k)$ to include the term $q_{i-k-1,i-k-1}$ at the previous step $(i-k)-1$. This matrix is of the right dimension to prove what we need. Now, using the recursive definition of the terms

$$q_{u,p}^{(i-k)} = -q_{i-k-1,i-k-1}^{(i-k-1)} \cdot q_{u,p}^{(i-k-1)} + q_{i-k-1,p}^{(i-k-1)} \cdot q_{u,i-k-1}^{(i-k-1)}$$

we substitute all the terms of the step $(i-k)$. We expand this to the whole matrix

$$\begin{vmatrix} q_{i-k-1,i-k-1}^{(i-k-1)} & \star & \dots & \star \\ 0 & q_{i-k,i-k}^{(i-k)} & \dots & q_{i-k,i}^{(i-k)} \\ 0 & \dots & \dots & \dots \\ 0 & q_{i,i-k}^{(i-k)} & \dots & q_{i,i}^{(i-k)} \end{vmatrix} =$$

$$= \begin{vmatrix} q_{i-k-1,i-k-1}^{(i-k-1)} & \star & \dots & \star \\ 0 & -q_{i-k-1,i-k-1}^{(i-k-1)} \cdot q_{i-k,i-k}^{(i-k-1)} + q_{i-k-1,i-k}^{(i-k-1)} \cdot q_{i-k,i-k-1}^{(i-k-1)} & \dots & \dots \\ 0 & \dots & \dots & \dots \\ 0 & -q_{i-k-1,i-k-1}^{(i-k-1)} \cdot q_{i,i-k}^{(i-k-1)} + q_{i-k-1,i-k}^{(i-k-1)} \cdot q_{i,i-k-1}^{(i-k-1)} & \dots & \dots \end{vmatrix}$$

The matrix now features only terms of the $(i-k-1)$ step. We want to write the matrix above as the product of the minor we are looking for and another support matrix. To write this matrix, notice that each term, regardless of its position is multiplied by $-q_{i-k-1,i-k-1}^{(i-k-1)}$, thus the idea of placing it on the main diagonal. Then, at each spot $_{u,p}$ we need to add the first of the uth row multiplied by the one above it (same pth column) in the first interesting row $(i-k-1)$. Hence in the first column of the first matrix (which is what will multiply for the element above you in your column), we place the elements of the first row. We extract the minor at the previous step as a product of matrices

$$= \begin{vmatrix} 1 & 0 & \dots & 0 \\ q_{i-k,i-k-1}^{(i-k-1)} & -q_{i-k-1,i-k-1}^{(i-k-1)} & \dots & 0 \\ q_{i-k+1,i-k-1}^{(i-k-1)} & 0 & -q_{i-k-1,i-k-1}^{(i-k-1)} & 0 \\ \dots & \dots & \dots & 0 \\ q_{i,i-k-1}^{(i-k-1)} & 0 & \dots & -q_{i-k-1,i-k-1}^{(i-k-1)} \end{vmatrix} \cdot \begin{vmatrix} q_{i-k-1,i-k-1} & \dots & \dots & q_{i-k-1,i} \\ q_{i-k,i-k-1} & \dots & \dots & q_{i-k-1,i} \\ \dots & \dots & \dots & \dots \\ q_{i,i-k-1} & \dots & \dots & q_{i,i} \end{vmatrix}^{(i-k-1)} =$$

Where the second matrix is what we are looking for, while for the first it is easy to compute the determinant (we are accounting for the fraction, which we neglected in the previous step)

$$\begin{vmatrix} q_{i-k,i-k}^{(i-k)} & \dots & q_{i-k,i}^{(i-k)} \\ \dots & \dots & \dots \\ q_{i,i-k}^{(i-k)} & \dots & q_{i,i}^{(i-k)} \end{vmatrix} = \frac{\left(-q_{i-k-1,i-k-1}^{(i-k-1)}\right)^{k-1}}{q_{i-k-1,i-k-1}^{(i-k-1)}} \cdot \begin{vmatrix} q_{i-k-1,i-k-1} & \dots & \dots & q_{i-k-1,i} \\ q_{i-k,i-k-1} & \dots & \dots & q_{i-k-1,i} \\ \dots & \dots & \dots & \dots \\ q_{i,i-k-1} & \dots & \dots & q_{i,i} \end{vmatrix}^{(i-k-1)}$$

Note that $k-1$ is an actual exponent! Since we are mostly interested in the sign, we write

$$sgn(\det(Q_{i-k,i}^{(i-k)})) = (-1)^{k-1} \left(q_{i-k-1,i-k-1}^{(i-k-1)}\right)^{k-2} sgn(\det(Q_{i-k-1,i}^{(i-k-1)}))$$

However, since all $q_{i,i}^{(i)}$ are negative, the scalar term is always negative. In fact , when k is even, the first $(-1)^{k-1} = -1$, while is k is odd, $(q_{i,i}^{(i)})^{k-2}$ is negative. Hence we have proven the inductive step.
At each step, we switch sign. To go from the ith step to the initial one we require $i-1$ step, and we can finally state that

$$sng(q_{i,i}^{(i)}) = (-1)^{i+1} \cdot sgn \left(\begin{vmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,i} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,i} \\ & \cdots & \cdots & \\ q_{i,1} & q_{i,2} & \cdots & q_{i,i} \end{vmatrix} \right) \tag{5.66}$$

Remember that we are asking for all $q_{i,i}^{(i)}$ to be negative, therefore the minors along the diagonal should have alternating signs.

$$sgn(q_{1,1}) = -1 \quad sgn \left(\begin{vmatrix} q_{1,1} & q_{1,2} \\ q_{2,1} & q_{2,2} \end{vmatrix} \right) = 1 \quad \ldots \quad sgn \left(\begin{vmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,i} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,i} \\ & \cdots & \cdots & \\ q_{i,1} & q_{i,2} & \cdots & q_{i,i} \end{vmatrix} \right) = -1^i$$

5.6.1 Lemmas used in the second model

Lemma 14. If η^{-1} has all its components non negative (non negative matrix) and $M(\vec{X}$ is diagonal and positive,

$$\max_t([\eta]^{-1} \cdot [M(\vec{X})] \cdot \vec{r}(t)) \leq [\eta]^{-1} \cdot [M(\vec{X})] \cdot \max_t(\vec{r}(t))$$

Lemma 15. Another useful thing to notice is that

$$\max_{x \in X} \{f(x)\} - \max_{x \in X} \{g(x)\} \leq \max_{x \in X} \{f(x) - g(x)\}$$

Proof.

$$\max_{x \in X} \{f(x)\} = \max_{x \in X} \{g(x) + f(x) - g(x)\} \leq \max_{x \in X} \{g(x)\} + \max_{x \in X} \{f(x) - g(x)\}$$

Where the last passage is true because we know that

$$\max_{x \in X} \{a(x) + b(x)\} \leq \max_{x \in X} \{a(x)\} + \max_{x \in X} \{b(x)\} \tag{5.67}$$

□

Bibliography

[1] L A Schmit. Structural Design by Systematic Synthesis. In 2nd Conference on Electronic Computation, pages 105-132, New York, NY, 1960. ASCE.

[2] Lucien A Schmit and William A Thornton. Synthesis of an Airfoil at Supersonic Mach Number. Technical Report CR 144, NASA, January 1965.

[3] L A Schmit Jr. Structural Synthesis âĂŤ Precursor and Catalyst. Recent Experiences in Multidisciplinary Analysis and Optimization. Technical Report CP-2337, NASA, 1984.

[4] Raphael T Haftka. Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements. Technical Report TN D-7264, NASA Langley Research Center,Hampton, VA, 1973

[5] Raphael T. Haftka. Optimization of flexible wing structures subject to strength and induced drag constraints. AIAA Journal, 14(8):1106-1977, 1977. doi:10.2514/3.7400.

[6] Raphael T. Haftka and C. P. Shore. Approximate methods for combined thermal/structural design. Technical Report TP-1428, NASA, June 1979.

[7] B Grossman, Z Gurdal, G J Strauch, W M Eppard, and R T Haftka. Integrated Aerodynamic/Structural Design of a Sailplane Wing. Journal of Aircraft, 25(9):855-860, 1988. doi:10.2514/3.45670.

[8] B. Grossman, R. T. Haftka, P.-J. Kao, D. M. Polen, and M. Rais-Rohani. Integrated aerodynamic-structural design of a transport wing. Journal of Aircraft, 27(12):1050-1056,1990.

[9] E. Livne, L.A. Schmit, and P.P. Friedmann. Towards integrated multidisciplinary synthesis of actively controlled fiber composite wings. Journal of Aircraft, 27(12):979-992, December 1990. doi:10.2514/3.45972.

[10] Eli Livne. Integrated aeroservoelastic optimization: Status and direction. Journal of Aircraft, 36(1):122 145, 1999.

[11] Evin J Cramer, J E Dennis Jr., Paul D Frank, Robert Michael Lewis, and Gregory R Shubin. Problem Formulation for Multidisciplinary Optimization. SIAM Journal on Optimization, 4(4):754-776, 1994. doi:10.1137/0804044.

[12] Raphael T Haftka. Simultaneous Analysis and Design. AIAA Journal, 23(7):1099 1103, 1985. doi:10.2514/3.9043.

[13] Natalia M Alexandrov and Robert Michael Lewis. Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design. AIAA Journal, 40(2):301-309, 2002. doi:10.2514/2.1646.

[14] Ilan M Kroo. MDO for large-scale design. In Multidisciplinary Design Optimization: State-of-the-Art, pages 22-44. SIAM, 1997.

[15] Richard J Balling and Jaroslaw Sobieszczanski-Sobieski. Optimization of Coupled Systems: A Critical Overview of Approaches. AIAA Journal, 34(1):6-17, 1996. doi:10.2514/3.13015

[16] Ilan M Kroo, Steve Altus, Robert Braun, Peter Gage, and Ian Sobieski. Multidisciplinary Optimization Methods for Aircraft Preliminary Design. In 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1994.

[17] Valerie M Manning. Large-Scale Design of Supersonic Aircraft via Collaborative Optimization. PhD thesis, Stanford University, 1999.

[18] R Balling and M R Rawlings. Collaborative Optimization with Disciplinary Conceptual Design. Structural and Multidisciplinary Optimization, 20(3):232-241, 2000. doi:10.1007/s001580050151.

[19] R Choudhary, A Malkawi, and P Y Papalambros. Analytic Target Cascading in Simulation-Based Building Design. Automation in Construction, 14(4):551-568, 2005. doi:10.1016/j.autcon.2004.11.004.

[20] Philip Geyer. Component-Oriented Decomposition for Multidisciplinary Design Optimization in Building Design. Advanced Engineering Informatics, 23(1):12-31, 2009. doi:10.1016/j.aei.2008.06.008.

[21] R Enblom. Two-Level Numerical Optimization of Ride Comfort in Railway Vehicles. Journal of Rail and Rapid Transit, 220(1):1-11, 2006. doi:10.1243/095440905X33279.

[22] Yuping He and John McPhee. Multidisciplinary Optimization of Multibody Systems with Application to the Design of Rail Vehicles. Multibody System Dynamics, 14(2):111-135, 2005. doi:10.1007/s11044-005-4310-0.

[23] Benjamin Potsaid, Yves Bellouard, and John Ting-Yung Wen. A Multidisciplinary Design and Optimization Methodology for the Adaptive Scanning Optical Microscope (ASOM). Proceedings of the SPIE, 6289:62890L1-12, 2006. doi:10.1117/12.680450

[24] M Kokkolaras, L S Louca, G J Delagrammatikas, N F Michelena, Z S Filipi, P Y Papalambros, J L Stein, and D N Assanis. Simulation-Based Optimal Design of Heavy Trucks by Model-Based Decomposition: An Extensive Analytical Target Cascading Case Study. International Journal of Heavy Vehicle Systems, 11:403-433, 2004. doi:10.1504/IJHVS.2004.005456.

[25] Charles D McAllister and Timothy W Simpson. Multidisciplinary Robust Design Optimization of an Internal Combustion Engine. Journal of Mechanical Design, 125(1):124-130, 2003. doi:10.1115/1.1543978.

[26] Rajesh Kalavalapally, Ravi Penmetsa, and Ramana Grandhi. Multidisciplinary optimization of a lightweight torpedo structure subjected to an underwater explosion. Finite Elements in Analysis and Design, 43(2):103-111, December 2006. doi:10.1016/j.finel.2006.07.005.

[27] Daniele Peri and Emilio F. Campana. Multidisciplinary design optimization of a naval surface combatant. Journal of Ship Research, 47(1):1-12, 2003.

[28] Ranjan Ganguli. Survey of recent developments in rotorcraft design optimization. Journal of Aircraft, 41(3):493-510, 2004.

[29] Bryan Glaz, Peretz P Friedmann, and Li Liu. Helicopter Vibration Reduction Throughout the Entire Flight Envelope Using Surrogate-Based Optimization. Journal of the American Helicopter Society, 54:12007-1-15, 2009. doi:10.4050/JAHS.54.012007.

[30] R D Braun, A A Moore, and I M Kroo. Collaborative Approach to Launch Vehicle Design. Journal of Spacecraft and Rockets, 34(4):478-486, 1997. doi:10.2514/2.3237

[31] Guobiao Cai, Jie Fang, Yuntao Zheng, Xiaoyan Tong, Jun Chen, and Jue Wang. Optimization of System Parameters for Liquid Rocket Engines with Gas-Generator Cycles. Journal of Propulsion and Power, 26(1):113-119, 2010. doi:10.2514/1.40649.

Chapter 6

A CubeSat Application

In this section we collapse the abstraction used for the formal proofs into a concrete example. For the most part, we will simply assign meaningful names to the various elements of $\vec{m}, \vec{x}(t), r(t)$ and $\vec{X}$; this will let us use the results presented so far directly. In one case, motivated by needs of the application, we introduce a new type of $\mathbf{B}_1(\cdot)$ function [1], and show that it is compatible with the proof structure used in Lemma 9. This is used as a pretext to review the types of guarantees we have to provide to expand the basic model.

As there are many ways in which we can describe a subsystem, we will have to make some model choices and assumptions. The aim of this chapter is to show how we verify the hypothesis needed for the existence of a strong optimum during the model design phase. Hence, we will not address the computational implementation, which will be the topic of the next chapter.

We discuss the preliminary design of a cubesat for a specific mission. The desired output of the algorithm shall be a set of specifications which we can use to develop the subsystems.

Mission specifications To run the algorithm, we need to have a quite detailed model of the mission. We will assume all the following information to be accessible.

- Payload specifications; its mass, size and thermal operational range. For each of relevant machine-states (such as payload off, idle, on, post processing etc), we need to have values for power consumption, data production, computational burden, etc

- Payload desired behavior; a time based schedule for the various machine-states of the payload

- Orbit details for the complete duration of the mission, including orbital maneuvers if present

- List of all the ground stations which we expect to contact

- Time based schedule of the desired spacecraft attitude at all time

- Form factor of the cubesat (6U vs 3U vs 1U)

[1] Which, we recall, performs the task of extracting the relevant requirement from a signal, like the maximum value.

Some of these information might come from an explicit requirement, like the payload specifications, while others might be only inferred from high level mission objectives. An example of the latter case might be the identification of the available ground stations. If we can choose freely, we can test multiple scenarios and determine which proves to be the best alternative.

There might be cases however, in which there is considerable uncertainty and we have no agency in the choice. For CubeSats, the chief example of this paradigm are the orbital parameters, which might not be fully defined at the time of design and which can not be chosen by the designer. As smallsat often rely on ride share capabilities, the final orbit is determined by the launcher and thus may remain unknown until well into the AIT phase. Therefore it is good practice to design for multiple possible orbit/scenarios. This can be implemented by extending the definition of mission to a sequence of possible scenarios. The outcome will be the best design, where the comparison is limited to the designs which are able to perform all possible mission scenarios.
Note that when applying this criteria, internal state parameters like battery energy levels, fuel in the tanks etc must be re-set to the initial values for each new scenario. This is allowed and can be modeled as an external supply signal $\vec{\Phi}(t)$, as was shown in the previous chapter.

Design specifications The algorithm will end with a design represented by a set of system parameters and a set of subsystem specifications. The level of details which we can obtain for the subsystem description depends on how accurately we choose to simulate their behavior. What follows is a list of the requirements which we can extract when modeling the components with simple building blocks:

Subsystem	Requirement	Type
ADCS	Maximum Torque	Authority
	Maximum angular momentum	Capacity
EPS	Maximum Current [Amp] rating per line	Authority
	Produced power	Authority
	Battery capacity	Capacity
Thermal Control	Heater sizing	Authority
Telecommunication	Radio datarate	Authority
On board computer	Computational performance	Authority
	Mass memory unit	Capacity
Propulsion	Thrust	Authority
	Tank sizing	Capacity

An interesting by-product of the algorithm is that the subsystem signals, like the instantaneous power consumption or the RW torque output, are available as time signals over the whole duration of the mission. This would allow subsystem designers to extract statistical information about how each component will be used, like average RPM or current or power consumption. In turn, these can be used to develop a better understanding of what will be requested of the subsystem and possibly better tailor its design.

6.1 Model for ACS

We are interested in the sizing of the actuators, in particular the RWs. For simplicity, we neglect the use of magneto rods. The subsystem requirements for the RW are to have sufficient authority, in terms of torque, and to prevent saturation, in terms of maximum angular momentum stored. Requirement on authority are the standard request we have modeled so far;

$$T_{\max,RW} \geq I_{\text{sat}} \cdot |\ddot{\theta}(t)| \quad \forall t \in [0, T_{\text{end}}] \quad \Rightarrow \quad \begin{cases} X_i &= & T_{\max,RW} \\ m_i(\vec{X}) &= & I_{\text{sat}} \\ r_i(t) &= & |\ddot{\theta}(t)| \\ x_i(t) &= & m_i(\vec{X}) \cdot r_i(t) \\ \mathbf{B}_{1,i}(\cdot) &= & \max_t(x_i(t)) \end{cases} \tag{6.1}$$

Where $\ddot{\theta}$ is the satellite angular acceleration as defined by mission specifications and I_{sat} models the inertia of the satellite along the same axis. This needs to be implemented for each of the RW in use, for example for $i = 1, 2, 3$.

On the other hand, RW capacity is compatible with maximum Δ in angular momentum

$$H_{\text{max,RW}} \geq \max_{t \in T} \Delta \left\{ \int_0^t I_{\text{sat}} \cdot \left(\ddot{\theta}^+(\tau) - \ddot{\theta}^-(\tau) \right) \mathrm{d}\tau \right\} \tag{6.2}$$

Where $\ddot{\theta}^+$ measures the clockwise angular acceleration and $\ddot{\theta}^-$ the counterclockwise ones. We can impose this by choosing the proper component of $\mathbf{B}_1$ to be $\max_{t \in T} \Delta$, constructing a $r_i(t)$ component as $\int_0^t \ddot{\theta}^+(\tau) - \ddot{\theta}^-(\tau) \mathrm{d}\tau$ and choosing a system parameter $m_i(X) = I_{\text{sat}}$. We have

$$X_i = H_{\text{max,RW}} = \mathbf{B}_1(x_i) = \max_{t \in T} \Delta(x_i(t)) = \max_{t \in T} \Delta(m_i(X) \cdot r_i(t)) = m_i(X) \cdot \max_{t \in T} \Delta(r_i(t)) \tag{6.3}$$

And the canonical mapping becomes

$$\begin{cases} X_i &= & H_{\max,RW} \\ m_i(\vec{X}) &= & I_{\text{sat}} \\ r_i(t) &= & \int_0^t \ddot{\theta}^+(\tau) - \ddot{\theta}^-(\tau) \mathrm{d}\tau \\ x_i(t) &= & m_i(\vec{X}) \cdot r_i(t) \\ \mathbf{B}_{1,i}(\cdot) &= & \max_t \Delta(x_i(t)) \end{cases} \tag{6.4}$$

6.2 Model for EPS

The requirements on the power system are multiple; to begin with, the EPS needs to be able to support the peak power. This parameter is essential in the choice of connectors and cables downstream of the EPS. The authority of the EPS is given by

$$P_{\text{max,EPS}} \geq \max_{t \in T} (x_{\text{EPS}}(t)) = \max_{t \in T} \left(\sum_{\text{SYS}=1}^{n} P_{\text{SYS}}(t) \right) = \max_{t \in T} \eta^{-1}|_i \cdot \mathbf{M}_{(\vec{X})} \cdot \vec{r}(t) \tag{6.5}$$

Where $\eta^{-1}|_i$ is the appropriate row of η^{-1} which extract the $x_{\text{EPS}}(t)$ component. The mapping from the canonical nomenclature used so far and the EPS specific one is thus

$$\begin{cases} X_i & = & P_{\text{max,EPS}} \\ m_i(\vec{X}) & = & 1 \\ r_i(t) & = & P_{\text{Payload}}(t) \\ x_i(t) & = & \eta^{-1}\mathbf{M}_{\vec{X}} \cdot \vec{r}(t) \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t(x_i(t)) \end{cases} \tag{6.6}$$

The integral of $x_{\text{EPS}}(t)$ on the other hand, provides the requirement for the power production system, the solar panels. The first assumption is certainly that the total amount of energy consumed by the system during the whole mission must be smaller than the energy produced

$$P_{\text{IN, tot}} \geq P_{\text{OUT, tot}} \quad \Rightarrow \quad P_{\text{IN, tot}} \geq \int_0^t x_{\text{EPS}}(\tau)\mathrm{d}\tau = \eta^{-1}|_i \cdot \mathbf{M}_{(\vec{X})} \cdot \int_0^t \vec{r}(\tau)\mathrm{d}\tau \tag{6.7}$$

Where $\eta^{-1}|_i$ is the appropriate row of η^{-1} which extract the $x_{\text{EPS}}(\tau)$ component. Again, the mapping to the canonical form for the solar panel array is

$$\begin{cases} X_i & = & P_{tot,\text{SP}} \\ m_i(\vec{X}) & = & {}^1/_{\varepsilon}|_{\text{EPS}} \\ r_i(t) & = & P_{\text{Payload}}(t) \\ x_i(t) & = & \eta^{-1}\mathbf{M}_{\vec{X}} \cdot \vec{r}(t) \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t(\int_0^t x_i(t)\mathrm{d}t) & = & \int_0^T x_i(t)\mathrm{d}t \end{cases} \tag{6.8}$$

Note that this specification is obtained from the same signal $x_i(t)$ which provided the authority for the EPS; in this case however we take the integral instead of the maximum value; the only difference is in the definition of $\mathbf{B}_{1,i}$. This highlight the fact that the number of requirements (which is the size of the requirement vector $\vec{X}$) does not have to be equal to the number of signals (which is the size of $\vec{x}$). However, if we use only linear model, the same condition can be expressed by doubling the number of signals and adding the $\int_0^T x_i(t)\mathrm{d}t$ as mission requirements, as it was done in the proof for primary storage.

Note that this requirement on power production is necessary but not at all sufficient. It is the battery capacity that drives the most stringent requirement for the solar panels, namely being able to compensate for the maximum excursion in the integral of net power. Then

$$\text{Cap}_{\text{batt}} \geq \max_{t \in T} \Delta \left(\int_0^t P_{\text{IN}}(\tau) - P_{\text{OUT}}(\tau)\mathrm{d}\tau \right) = \max_{t \in T} \Delta \left(\int_0^t x_{\text{Sol}}(\tau) - x_{\text{EPS}}(\tau)\mathrm{d}\tau \right) \tag{6.9}$$

This condition is new because it involves max Δ between two signals which are affected by system parameters. In particular, one could think that increasing solar panel area[2] would decrease the requirement on battery capacity. This would lead to possible decrease in overall mass, inertia and so on. All this would be against the basic assumptions for the main lemma.

The following section will be devoted to provide a formal assurance that, for most missions, this is not the case. Intuitively, we can observe that energy production and storage are not equivalent; no

[2] which we can see as a requirement X which is used directly as a system parameters m_i to multiply solar flux ϕ

matter how big the solar array is, during eclipse condition, a battery is needed. Similarly, for any mission that lasts more than a few hours, solar panels must be used. In general, we can expect the capacity of the battery to be driven by the worst condition encountered during the mission, which is when the solar panel do not work.

In order to apply Lemma 9, we need to prove that for the proposed requirement function $\mathbf{B}_1$

$$X_k = \mathbf{B}_1(x_i(t)) = \max_{t\in T} \Delta \left(\int_0^t x_{\text{Sol}}(\tau) - x_{\text{EPS}}(\tau) \mathrm{d}\tau \right) \tag{6.10}$$

It is still true that

$$\vec{y}(t) \ll \vec{z}(t) \quad \Rightarrow \quad \mathbf{B}_1(\vec{y}(t)) \ll = \mathbf{B}_1(\vec{z}(t)) \tag{6.11}$$

and

$$\|\mathbf{B}_1(\mathbf{M}_{(\vec{Y})} \cdot \vec{r}(t)) - \mathbf{B}_1(\mathbf{M}_{(\vec{Z})} \cdot \vec{r}(t))\| \leq \| \left(\mathbf{M}_{(\vec{Y})} - \mathbf{M}_{(\vec{Z})} \right) \cdot \mathbf{B}_1(\vec{r}(t)) \| \tag{6.12}$$

If both conditions hold, we can re-use the same proof structure of the Lemma 9.

First part We begin with condition 6.11. Neglecting all components of $\vec{y}(t), \vec{z}(t)$ which are not relevant, like the torque of the RW, the force outputted by the thrusters etc, the first inequality in 6.11 states

$$\vec{y}(t) \ll \vec{z}(t) \quad \Leftrightarrow \quad \int_0^t x_{\text{Sol}}(\tau)|_y - x_{\text{EPS}}(\tau)|_y \,\mathrm{d}\,\tau < \int_0^t x_{\text{Sol}}(\tau)|_z - x_{\text{EPS}}(\tau)|_z \,\mathrm{d}\,\tau \quad \forall t \in [0, T] \tag{6.13}$$

We must prove that this implies the same inequalities for the $\max \Delta(\cdot)$ of the functions. This is easy to prove if we assume that (for both signals y and z)

$$\int_0^t x_{\text{EPS}}(\tau) \mathrm{d}\tau \leq \int_0^t x_{\text{Sol}}(\tau) \mathrm{d}\tau \quad \forall t \in [0, T] \tag{6.14}$$

Then, since both are strictly increasing functions which have value zero at $t = 0$, and with a change of names to lighten the nomenclature

$$g(t) = \int_0^t x_{\text{EPS}}(\tau) \mathrm{d}\tau \qquad f(t) = \int_0^t x_{\text{Sol}}(\tau) \mathrm{d}\tau \tag{6.15}$$

$$\max_t \Delta(f(t) - g(t)) = \max_t (f(t) - g(t)) - \underbrace{\min(f(t) - g(t))}_{=0} = \max_t (f(t) - g(t)) \tag{6.16}$$

We have

$$f(t)|_y - g(t)|_y < f(t)|_z - g(t)|_z \quad \Rightarrow \quad \max_t (f(t)|_y - g(t)|_y) < \max_t (f(t)|_z - g(t)|_z) \tag{6.17}$$

Note that condition 6.14 is very reasonable and only mildly conservative; it requires that, at any given time, we can not have consumed more energy than the amount we have collected. This might not be strictly true in the first few minutes of operations, but it soon becomes sensible as the energy consumed exceeds the initial quantity stored in the battery.

Still, one might be curios on how we can guarantee condition 6.14. This needs to be done in the

$\mathbf{B}_2(\cdot)$ function where the design of the solar panel takes place. We need to propose only solutions for which this condition is verified. In this way, we have the full chain condition 5.16 to be true:

$$\vec{X}_1 \underset{=}{\ll} \vec{X}_2 \quad \Rightarrow \quad \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_1)}(\vec{r}(t))\right) \underset{=}{\ll} \mathbf{B}_1\left(\mathbf{A}_{\mathbf{B}_2(\vec{X}_2)}(\vec{r}(t)\right) \tag{6.18}$$

One way to practically enforce this is to use backtracking. We assume condition 6.14 to be true and, if during the next mission simulation we discover that it was not, we backtrack and propose a new design. We proceed only if the condition is satisfied. This provides a series of designs which all comply with hypothesis and lead to the strong optimum.

Second part We now move to the second proof, where we need to verify that it is still possible to separate requirement signals $r_i(t)$ from system parameters m_i, as requested by condition 6.12 (re-stated below)

$$\|\mathbf{B}_1(\mathbf{M}_{(\vec{Y})} \cdot \vec{r}(t)) - \mathbf{B}_1(\mathbf{M}_{(\vec{Z})} \cdot \vec{r}(t))\| \leq \|\left(\mathbf{M}_{(\vec{Y})} - \mathbf{M}_{(\vec{Z})}\right) \cdot \mathbf{B}_1(\vec{r}(t))\|$$

To simplify notation, initially we consider $\eta = \mathbb{I}$, which provides $x_{\text{EPS}}(t) = m_i \cdot r_i(t)$. We also consider that only one face is exposed to the sun, hence $x_{\text{Sol}}(t) = A_{\text{Sol}} \cdot \phi_{\text{sun}}(t)$ or $x_{\text{Sol}}(t) = m_j \cdot r_j(t)$. We focus only on the components of $\mathbf{B}_1$ in the form of 6.10. We define $f(t) \doteq \int_0^t r_i(\tau)\mathrm{d}\tau$ and $g(t) \doteq \int_0^t r_j(\tau)\mathrm{d}\tau$ to simplify notation. The left hand side of equation 6.12 can be written as

$$\| \max_{t\in T} \Delta\left(m_i|_Y f(t) - m_j|_Y g(t))\right) - \max_{t\in T} \Delta\left(m_i|_Z f(t) - m_j|_Z g(t))\right) \| \tag{6.19}$$

Which we can bound from above with

$$\leq \| \max_{t\in T} \Delta\Big(m_i|_Y f(t) - m_j|_Y g(t) - m_i|_Z f(t) + m_j|_Z g(t))\Big) \| \tag{6.20}$$

Now we can collect the common terms $f(t), g(t)$

$$= \| \max_{t\in T} \Delta\Big(f(t) \cdot (m_i|_Y - m_i|_Z) - g(t) \cdot (m_j|_Y - m_j|_Z)\Big) \| \tag{6.21}$$

Calling $\Delta m_i \doteq m_i|_Y - m_i|_Z$ and $\Delta m_j \doteq m_j|_Y - m_j|_Z$ and applying the definition of $\max\Delta$ we can write

$$= \| \max_{t\in T} \Delta\left(\Delta m_i f(t) - \Delta m_j g(t)\right) \| = \| \max_{t\in T}\left(\Delta m_i f(t) - \Delta m_j g(t)\right) - \min_{t\in T}\left(\Delta m_i f(t) - \Delta m_j g(t)\right) \| \tag{6.22}$$

Using the same notation, the right side of condition 6.12 is expressed as

$$\| \left(\mathbf{M}_{(\vec{Y})} - \mathbf{M}_{(\vec{Z})}\right) \cdot \mathbf{B}_1(\vec{r}(t))\| = \| |\Delta m_i| \max_{t\in T} \Delta(f(t)) - |\Delta m_j| \max_{t\in T} \Delta(g(t)) \| \tag{6.23}$$

To proceed, we consider the sign of Δm_i and Δm_j; there are 4 possible cases, but they are in pair equivalent. Recalling that both $f, g > 0$, we can have $\Delta m_i > 0$ and $\Delta m_j < 0$ (and the complementary $\Delta m_i < 0$ and $\Delta m_j < 0$) or $\Delta m_i > 0$ and $\Delta m_j > 0$ (and complementary $\Delta m_i < 0$ and $\Delta m_j < 0$).

First case, $\Delta m_i > 0$ and $\Delta m_j < 0$ We use absolute values for m_i and m_j to explicitly write the signs; a vertical disposition is used only to facilitate element comparison, intended in the horizontal direction

$$\left\| \begin{array}{ll} & \max_t \Big(|\Delta m_i| f(t) + |\Delta m_j| g(t)\Big) \\ - & \min_t \Big(|\Delta m_i| f(t) + |\Delta m_j| g(t)\Big) \end{array} \right\| \leq \left\| \begin{array}{llll} & \max_t \Big(|\Delta m_i| f(t)\Big) + & \max_t \Big(|\Delta m_j| g(t)\Big) \\ - & \min_t \Big(|\Delta m_i| f(t)\Big) - & \min_t \Big(|\Delta m_j| g(t)\Big) \end{array} \right\| = \tag{6.24}$$

Proof for the complementary case ($\Delta m_i < 0$ and $\Delta m_j > 0$) is identical, considering that, for $f, g > 0$, $\max(-f-g) = -\min(f+g)$ and $\min(-f-g) = -\max(f+g)$.

Second case, $\Delta m_i > 0$ and $\Delta m_j > 0$

$$\left\| \begin{array}{ll} & \max_t \Big(|\Delta m_i| f(t) - |\Delta m_j| g(t)\Big) \\ - & \min_t \Big(|\Delta m_i| f(t) - |\Delta m_j| g(t)\Big) \end{array} \right\| \leq \left\| \begin{array}{ll} \max_t \Big(|\Delta m_i| f(t)\Big) - & \min_t \Big(|\Delta m_j| g(t)\Big) \\ \max_t \Big(|\Delta m_i| g(t)\Big) - & \min_t \Big(|\Delta m_j| f(t)\Big) \end{array} \right\| = \tag{6.25}$$

For the complementary case, we invert the order, which leads to the same result when considering the absolute value $\| \cdot \|$.

General case, for multiple panels In the general case, solar power is produced with an distribution of solar cells on different faces; therefore, we write

$$\max_{t \in T} \left(+A_{\mathrm{x}} \int_0^t \phi_x(\tau) \mathrm{d}\tau + A_{\mathrm{y}} \int_0^t \phi_y(\tau) \mathrm{d}\tau + A_{\mathrm{z}} \int_0^t \phi_z(\tau) \mathrm{d}\tau - \int_0^t P_{\mathrm{out}}(\tau) \mathrm{d}\tau \right) \tag{6.26}$$

Which leads to

$$\leq \| \max_{t \in T} (\Delta m_0 f_0(t) + \Delta m_1 f_1(t) + \Delta m_2 f_2(t) - \Delta m_3 f_3(t) - \dots) \| = \| \max_{t \in T} \Big(\sum_i c_i \cdot f_i(t) \Big) - \min_{t \in T} \Big(\sum_i c_i \cdot f_i(t) \Big) \| \tag{6.27}$$

It is easy to see that, calling $\mathcal{I}$ the set of indices for which $c_i > 0$ and $\mathcal{J}$ the set of indices for which $c_i < 0$, the maximum of the difference will be smaller than the difference between maxima and minima

$$\max_{t \in T} \left(\sum_i c_i \cdot f_i(t) \right) \leq \max_{t \in T} \underbrace{\left(\sum_i^{\mathcal{I}} |c_i| \cdot f_i(t) \right)}_{\text{for } c_i > 0} - \min_{t \in T} \underbrace{\left(\sum_j^{\mathcal{J}} |c_j| \cdot f_j(t) \right)}_{\text{for } c_j < 0} \tag{6.28}$$

and that, conversely, the minimum of the difference will be greater than the difference between minima and maxima

$$\min_{t \in T} \left(\sum_i c_i \cdot f_i(t) \right) \geq \min_{t \in T} \underbrace{\left(\sum_j^{\mathcal{I}} |c_j| \cdot f_j(t) \right)}_{\text{for } c_j > 0} - \max_{t \in T} \underbrace{\left(\sum_i^{\mathcal{J}} |c_i| \cdot f_i(t) \right)}_{\text{for } c_i < 0} \tag{6.29}$$

which is simply the generalization of the case with only two signals. Inverting the sign of the second equation and taking the sum of both we have

$$\begin{array}{ccc} \max_{t\in T}\left(\sum_i c_i\cdot f_i(t)\right) & & \max_{t\in T}\underbrace{\left(\sum_i^{\mathcal{I}}|c_i|\cdot f_i(t)\right)}_{\text{for } c_i>0} - \min_{t\in T}\underbrace{\left(\sum_j^{\mathcal{J}}|c_j|\cdot f_j(t)\right)}_{\text{for } c_j<0} \\ + & \leq & + \\ -\min_{t\in T}\left(\sum_i c_i\cdot f_i(t)\right) & & -\min_{t\in T}\underbrace{\left(\sum_j^{\mathcal{I}}|c_j|\cdot f_j(t)\right)}_{\text{for } c_j>0} + \max_{t\in T}\underbrace{\left(\sum_i^{\mathcal{J}}|c_i|\cdot f_i(t)\right)}_{\text{for } c_i<0} \end{array} \tag{6.30}$$

Which finally leads us to

$$\| \max_{t\in T}\left(\sum_i c_i\cdot f_i(t)\right) - \min_{t\in T}\left(\sum_i c_i\cdot f_i(t)\right)\| \leq \|\sum_i c_i\cdot \max\Delta(f_i(t))\| \tag{6.31}$$

6.3 Model for Thermal Control

The thermal model for a satellite can become very complex. Here we only present a simple, single node model that could be used to gain order of magnitude information about the heaters output and to account for their impact on the power budget. Knowing the satellite form factor, its optical properties, and an acceptable temperature profile, we can estimate net heat exchanged with the environment.

$$q_{\text{net, env}}(t)\doteq q_{\text{in}} - q_{\text{out}}(T_{\text{wanted}}(t)) \tag{6.32}$$

We assume that the spacecraft is designed following a cold bias approach, meaning that the active part of the thermal control system can only add heat. This is a forced choice in small satellites, where due to power budget limitations, active heat removal is unfeasible. Then, the role of the heater is to provide sufficient heat to keep internal temperature within acceptable range during the coldest possible condition. Its contribution can be modeled as

$$q_{\text{heater}}(t) = c_m m_{\text{sat}}\cdot\frac{\mathrm{d}}{\mathrm{d}t}T_{\text{wanted}}(t) - q_{\text{net, env}}(t) \tag{6.33}$$

The canonical map can be written considering the heat generated by all other subsystems using the matrix η,

$$\begin{cases} X_i & = & \mathrm{q}_{\text{max,heater}} \\ m_i(\vec{X}) & = & c_m m_{\text{sat}} \\ r_i(t) & = & {}^{\mathrm{d}}/_{\mathrm{d}t}(T_{\text{wanted}}) \\ \Phi_i(t) & = & -q_{\text{net, env}} \\ x_i(t) & = & m_i\cdot r_i(t) + \Phi(t) \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t(x_i(t)) \end{cases} \tag{6.34}$$

It is to be noted that $q_{\text{in}}(t)$ is not obtained by the waste heat signal of the other subsystem; instead it is the minimum heat generated internally by the spacecraft. There are two reasons for this; the first is that, due to the model assumptions, we can not allow for an increase of a system parameter (like the moment of inertia) to decrease the requirement on the heaters (for example due to an increase in waste heat of the RW). The second is more due to a conservative philosophy; in the mission scenario we should include a period of worst cold case in which all but the essential subsystems are powered off. The heater should be sized to support the survival of the craft under such conditions.

6.4 Model for On Board Computer

The model for the on board computer (OBC) will inevitably be highly simplified. Our goal is merely to capture the impact it has on the power budget and, conversely, the burden that other subsystems might have on its workload. As an example of an additional load caused on another subsystem, we can imagine that some post processing might be required to compress the data before downlink. Hence the payload(or the radio) might require the OBC to perform some computation during specified time frames. We want to make sure that the OBC is able to sustain the throughput for the whole mission duration as well as accounting for its impact on the power system.

A proper characterization of a computational system is a non trivial task; power-consumption-per-task are hard to assess from specifications and time consuming to generate with experiments. Therefore we opt for a simplified model in which the output signal linearly maps to power consumption, total heat generated and computational throughput.

We imagine the state variable as number of operation per second, not to dissimilar from the clock rate. As more tasks are added in the OBC queue, the number of operation per Δt increases and so does power consumption and generated heat. Realistically, the processor will not use CPU throttling as intensively as our model would suggest; assuming the OBC is a soft real time system, it would simply perform all the requests in the queue and then wait until new tasks are available. This behavior will, on average, appear identical to our model.

Then, we specify the requirements of the OBC as maximum clock rate and maximum storage space. The first can be obtained with an authority model, while the second with a capacity model based on the integral of the difference between produced and downloaded data.

$$\mathrm{Cl}_{\mathrm{max,OBC}} \geq \max_{t\in T}\left(\mathrm{Cl}_{\mathrm{OBC}}(t)\right) = \max_{t\in T}\left(\sum_{\mathrm{SYS}=1}^{n} \mathrm{Cl}_{\mathrm{SYS}}(t)\right) = \max_{t\in T}\left(\eta^{-1}|_i \cdot \mathbf{M}_{(\vec{X})} \cdot \vec{r}(t)\right) \tag{6.35}$$

The mapping to the canonical is

$$\begin{cases} X_i & = & \mathrm{Cl}_{\mathrm{max,OBC}} \\ m_i(\vec{X}) & = & 1 \\ r_i(t) & = & \mathrm{Cl}_{\mathrm{House\ keeping}} \\ x_i(t) & = & \eta^{-1}\mathbf{M}_{\vec{X}} \cdot \vec{r}(t) \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t(x_i(t)) \end{cases} \tag{6.36}$$

For the mass memory storage unit (MMU), we can simply integrate net data flow

$$\mathrm{Cap}_{\mathrm{MMU}} \geq \max_{t\in T} \Delta \left(\int_0^t D_{\mathrm{downlink}}(\tau) - D_{\mathrm{produced}}(\tau)\mathrm{d}\tau\right) \tag{6.37}$$

If data production is given by mission requirement, hence it is independent of the subsystem signals $x_j(t)$, we can treat it as a known contribution $\Phi(t)$. Alternatively, if subsystems behavior significantly affects data production, we need to use a scheme similar to the one used to set the capacity of the battery unit.
In the latter case, we need to ask for the total amount of data downloaded at any time t to be smaller than the total amount of data generated.

$$\int_0^t D_{\text{downlink}}(\tau)\mathrm{d}\tau \geq \int_0^t D_{\text{produced}}(\tau)\mathrm{d}\tau \qquad \forall t \in [0, T] \tag{6.38}$$

This is a rather reasonable assumption, and must be verified during the algorithm implementation, as for the solar panel. The canonical mapping is

$$\begin{cases} X_i & = & \text{Cap}_{\text{MMU}} \\ m_i(\vec{X}) & = & 1 \\ r_i(t) & = & \int_0^t D_{\text{downlink}}(\tau) - D_{\text{produced}}(\tau)\mathrm{d}\tau \\ x_i(t) & = & m_i \cdot r_i(t) \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t \Delta(\cdot) \end{cases} \tag{6.39}$$

6.5 Model for Telecommunication

We want to determine the minimum datarate required by the mission. The amount of data to be transmitted typically depends on the payload characteristics and mission specifications. Additionally, subsystems may contribute with housekeeping data which need to be downloaded for diagnostics and system health monitoring. As we can only download data when we are above a ground station, we capture the intermittent nature of link availability with the function GS(t), which takes value 1 when the satellite is in view of a ground station and 0 otherwise. Therefore, the requirement on datarate is

$$\text{Dr} \geq \frac{\int_0^{t_{\text{end}}} D|_{\text{payload}}(\tau) + \sum_i^n D|_{\text{subs i}}(\tau)\ \mathrm{d}\tau}{\int_0^{t_{\text{end}}} \text{GS}(\tau)\mathrm{d}\tau} \qquad \left[\frac{\text{Mb}}{\text{s}}\right] \tag{6.40}$$

Similarly to requirements for storage systems, the requirement for the radio does not depend on its output signal $x(t)$. However, we still want to consider it to track its effect on power budget, which is typically not negligible. We construct the signal $x(t)$, the link Mbps, by turning on the radio when there is data to download and the ground station is visible, turning it off otherwise.

$$x(t) = \begin{cases} \text{GS}(t) \cdot \text{Dr} \quad \text{if} \quad \int_0^t D|_{\text{payload}}(\tau) - x(\tau)\ \mathrm{d}\tau > 0 \\ \\ \quad 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{6.41}$$

The map to canonical form for the radio is

$$\begin{cases} X_i & = & \text{Dr} \\ m_i(\vec{X}) & = & 1 \\ r_i(t) & = & \text{GS}(t)^* \\ x_i(t) & = & m_i \cdot r_i(t) \cdot \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t(x_i(t)) \end{cases} \tag{6.42}$$

Where GS* is ground station availability modified according to download necessity.

6.6 Model for Propulsion

The propulsion system can be modeled using the basic blocks since we are only interested in maximum authority and propellant quantity. Since we are considering CubeSats and it is a simplified case, we neglect change of mass during firing. This assumption is consistent with the following scenarios:

- We are using a one shot chemical engine. The burn time is short, so we might just use a suitably averaged mass for the system and consider it constant
- We are using electric propulsion with sufficiently high specific impulse such that the change of mass is overall negligible

The notable case which is outside these scenarios is the use of cold gas thrusters, which have extremely low specific impulse and can be used multiple times. Unfortunately for our model, these are the most popular form of propulsion system for cubesats. Model that account for time varying system parameters would be an interesting extension for this framework. A crude way to deal with this limitation would be to artificially modify the ΔV request signal, reducing it to account for mass reduction.

$$\mathrm{Th}_{\max} \geq \max_t(m_{\mathrm{sat}} \cdot \frac{|\Delta \mathrm{V}(t)|}{\Delta t}) \quad \forall t \in [0, T_{\mathrm{end}}] \qquad \begin{cases} X_i & = & \mathrm{Th}_{\max} \\ m_i(\vec{X}) & = & m_{\mathrm{sat}} \\ r_i(t) & = & |\Delta \mathrm{V}(t)| \\ x_i(t) & = & m_i(\vec{X}) \cdot r_i(t) \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t(x_i(t)) \end{cases} \tag{6.43}$$

As for the RWs this model would be implemented multiple times, according to the number of thrusters in the CubeSat.

Assuming that there is a single fuel tank, and that no resupply is scheduled, the sizing of the tank can be reduced to a simple primary capacity model

$$M_{\mathrm{tot,fuel}} \geq M_{\mathrm{used}} = \int_0^t \sum_{i=1}^n \dot{m}_i(\tau)\mathrm{d}\tau = \int_0^t \sum_{i=1}^n \frac{F_i(\tau)}{g_0 \cdot \mathrm{Isp}}\mathrm{d}\tau = \frac{m_{\mathrm{sat}}}{g_0 \cdot \mathrm{Isp}} \int_0^t \sum_{i=1}^n \Delta \mathrm{V}_i(\tau)\mathrm{d}\tau \tag{6.44}$$

The mapping to the canonical form for the fuel tank is

$$\begin{cases} X_i & = & M_{\mathrm{tot,fuel}} \\ m_i(\vec{X}) & = & m_{\mathrm{sat}} \\ r_i(t) & = & \Delta \mathrm{V}_i(\tau) \\ x_i(t) & = & \sum_{j=1}^n m_j \cdot r_j(t) \\ \mathbf{B}_{1,i}(\cdot) & = & \max_t \left(\frac{m_{\mathrm{sat}}}{g_0 \cdot \mathrm{Isp}} \cdot \int_0^t x_i(t)\mathrm{d}t \right) \quad = \quad \frac{m_{\mathrm{sat}}}{g_0 \cdot \mathrm{Isp}} \cdot \int_0^T x_i(t)\mathrm{d}t \end{cases} \tag{6.45}$$

Chapter 7

An Example Implementation

In the previous chapters we showed a specific mathematical model which featured some appealing properties for system design, namely existence and uniqueness of a robustly optimal solution. In this chapter we present a minimalistic version of the algorithm to assess its merits in a concrete application, the design of a CubeSat. The hope is that, by discussing a computational implementation, we will clarify some of the details which were overlooked in the abstract model, as for example how to pursue the *design for requirements* function in non-trivial cases.

The software will be tested using a recent mission as a target and the optimality of the proposed design corroborated numerically by exploring randomly selected neighboring points in the design space. The model is limited to only power budget and attitude as it is a demonstrator and a full system design would be beyond the scope of this prototype.

7.1 Overview of the software

At the conceptual level, the algorithm is simply an iterative process, as shown in Fig 7.1; requirements are derived assuming some parameters, a system is designed to meet those requirements, its parameters are used to derive new requirements and the process repeats.

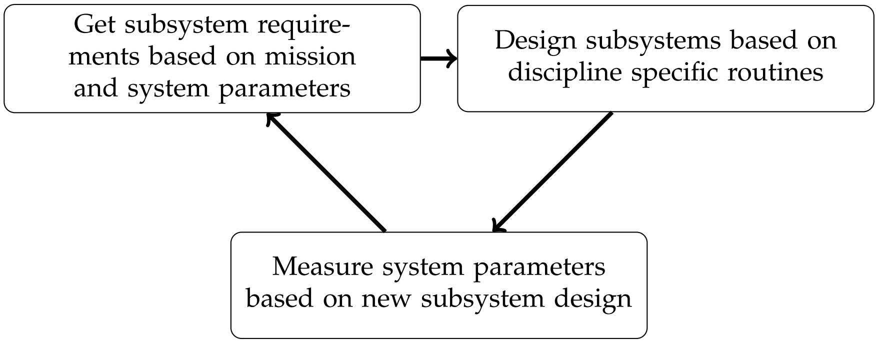

Get subsystem requirements based on mission and system parameters
Design subsystems based on discipline specific routines
Measure system parameters based on new subsystem design

Figure 7.1: The basic iterative process; we obtain requirements based on mission description and some assumptions, we then design each subsystem and finally update initial assumptions.

In the first step, requirements are extracted from the *signals* which the system must be able to output to accomplish the given objective. Such signals are obtained by simulating the whole mission and recording subsystem inputs, what they require to operate (like current and fuel) and outputs, which might adversely impact other subsystems (like waste heat, data to be stored etc). Hence, the first block requires a complete mission definition, a simulation tool able to reproduce the environmental conditions (like position along the orbit, direction of sunlight, thermal loads etc) and a model of the system, which reacts to external conditions with the signal needed to meet mission requirement.

The second part of the software models the design of each subsystem with the aim of fulfilling requirements. This has been deliberately overlooked in the previous chapter as it is dependent on the specific subsystem. For some, it is trivial; if the peak output power is 12W, the EPS must be able to output at least 12W. For others, there is room for optimization; multiple solar cells distributions can be used to generate similarly valid power profile, which one to choose is left to the design procedure.

The final step is to evaluate system parameters, such as mass, inertia etc, from a given design. While some parameters can be scaled linearly, like the mass of the battery pack with the number of battery cells, some are intrinsically not linear, such as the moment of inertia, which depends on internal mass distribution.

Notably, any design strategy which satisfies the hypothesis of Lemma 9 will lead to *a* strong optimum configuration. Clearly, different strategies will lead to different optima, as we are essentially changing the technology used for the system. However it should be easy to identify which is the best strategy for each domain; again in the case of the solar cell distribution, it is intuitive to prefer the configuration which uses the smallest number of solar cells.

Notes on dependencies
The prototype is written in Python 3 for convenience. For numerical computation it relies on NumPy, for graphical outputs it uses matplotlib, for orbit propagation it relies on poliastro.

7.1.1 Mission definition

A rather detailed definition of the mission is needed to faithfully simulate the system behavior. From the software point of view, we wish to know what is requested of each subsystem at each instant of time; we want to know when the payload will be turned on and consuming power/producing data to be stored and re-transmitted, we want to know when we will be in view of the ground station and what attitude we need to maintain the link and we also want to know the torques that the ADCS must provide.

As it is unpractical for the user to hard code such low level specifications, the software accepts a more abstract definition. The user can define the mission by specifying the orbit (from classical parameters) and then select which ground target needs to be followed by the various subsystems, depending on a few logical rules. For example, a ground station can be modeled as a point on the surface of the earth; the satellite is instructed to point a specific antenna(described as a direction in the body frame) toward the station only when inside the field of view of the target. Similarly, it is possible to mark a ground area as interesting for an earth observation payload. Then as the satellite passes over it, it shall switch on the instrument and orient its bore-sight toward nadir.

Using these specifications it is possible to easily account for all the necessary subsystem interactions; payload operations affect the EPS as they increases power demand but also impose strict demand on the ADCS which, orienting the system toward the target, also contributes to power demand. By changing the attitude of the spacecraft, the electrical power produced by the solar panels may change significantly, depending on the solar cell distribution, and further complicating the problem.

It is also important to notice that, although CubeSats were introduced as demonstrators, they are often used in very ambitious missions with high duty cycles. This renders the timing of the scheduled operations more critical. For example, as the downloading of payload data is a typically power hungry operation, performing it during eclipse or in daylight has a very different impact on the power budget. To assess and control the impact of these choices, a logic layer can be used to specify when to perform any given operation. For example, boolean operators can be used to indicate that a radio link needs to be established whenever in view of a selected ground station but only if during daylight. Similarly, it is possible to assign higher priority to the payload by forbidding the scheduling of ground station communication if payload operation are also possible at the same time. Finally, to account for subsystem specific time requirements, it is possible to specify temporal extension of operations. For example, to give the ADCS enough time to initiate a tracking maneuver, GS pointing can be scheduled to begin some minutes before it enters the field of view of the spacecraft. In much the same way, a subsystem may require a boot or warm-up period before coming online or a processing/shut down time after its operations. It is very easy to extend the time window in which it is operational; in this way we can more faithfully account for its impact on the power budget.

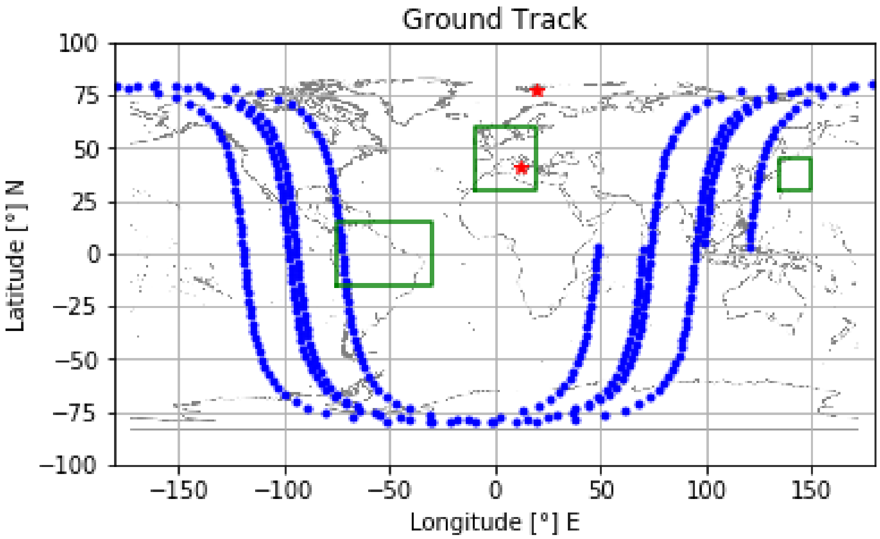


Figure 7.2: An example of mission definition, with ground track in blue, selected ground stations in red and areas of interest in green.

7.1.2 Subsystem design

The basic model for the spacecraft is implemented as a class which coordinates the subsystems in three phases; at first during the simulation, then during the design and finally during the system parameters evaluation.
During the simulation, it is tasked with collecting the various contributions and assigning them to the appropriate subsystem; for example, it gathers the various power consumption from the subsystems and assigns their sum to the EPS. During the design phase, it triggers the subsystem specific redesign procedure ensuring the correct requirements (maximum torque, power demand etc) are available, which typically are completely known only at the system level. In the last phase, it uses the predictions of each subsystem to estimate the global parameters; for example, after having the masses of each component, a value for the moment of inertia of the spacecraft can be derived. This is also helped by the standard CubeSat format, which fixes the size of the satellite.

The most interesting part to review is the design of the EPS, in particular how to place the solar cells on the various faces of the CubeSat.

Providing a solar cell distribution

The goal for the solar panel distribution is to guarantee satisfaction of condition 7.1, namely that, at any given time, produced energy is greater than the consumed energy. There is more than one configuration that can meet this requirement.

$$\int_0^t P_{\text{solar}}(\tau)\mathrm{d}\tau \geq \int_0^t P_{\text{out}}(\tau)\mathrm{d}\tau \quad \forall t \in [0, T_{\text{end}}] \tag{7.1}$$

The problem can be recognized as well posed, as we can accept that it is beneficial to choose the distribution with the lowest number of cells, so the cost function in this optimization is unambiguously, although implicitly, defined. Furthermore, we know that either the problem has a solution or we can prove that it does not by using the maximum number of cells on each face and verify that it still is not enough. Finally it is reasonable to assume that if multiple options exist with exactly the same number of cells, we are indifferent to which one we choose.

We will now present an algorithm that produces an acceptable solution to the problem as posed above.

When dealing with a 6U cubesat, we could naively think that, since the possible combinations are finite (as each face has a maximum number of cells) we could simply try them all. Assuming that a 6U face can house 14 cells, a 3U face up to 7, and a 2U up to 4, the total number of possible combinations is $14 \cdot 14 \cdot 7 \cdot 7 \cdot 4 \cdot 4 \approx 10^5$. Considering that, to test whether a solution complies with the constraint 7.1 we potentially need to simulate the whole mission up to the time T_{end} which could be some months, and possibly many times over if there is uncertainty on which orbit we will use, this option is clearly very expensive.

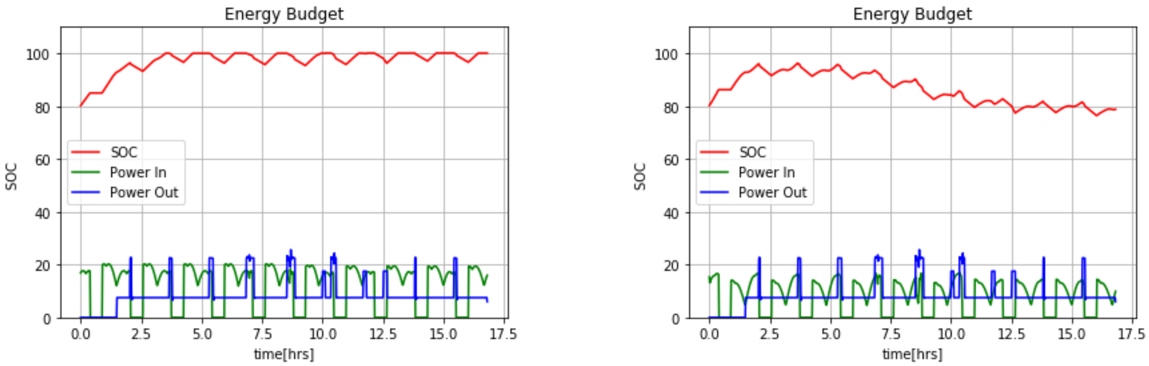
A first approximate solution could be to note that, for specific orbits, there is low variability with regard to the sunlight. This is the case of sun synchronous orbits. Then a good strategy might be to increase the number of cells on the most effective face as needed[1]. Once this is maxed out, if more energy is required, we move to increase the second best and so on.

[1] The most effective face being the one that receives more energy on average over the whole mission.

This approach is very straight forward, but it is important to note that it might fail spectacularly in particular cases. If the *best* face is obscured for the first part of the mission, no number of cells will make the power budget positive. Worse yet, if a minimal amount of sunlight reaches the "average best" face, during a critical period, then a disproportionate amount of cells will be used and the solution will be perceived as best, while much better options could exist.

To prevent this potential pitfalls, the software also implements a genetic optimization strategy and then compares the two results. If both are able to meet requirement 7.1, the one with the smaller number of cells is chosen.

Note that, even though constraint 7.1 is very reasonable, it might be too demanding in the initial phase of the mission. If the simulation happens to begin while the system is in eclipse, the problem can not be solved. To avoid these technical problems we can either make sure that the system turns on while in sunlight (thus avoiding any power consumption while in eclipse) or we can set the solution to allow violation of the constraint up to the time t. We implemented the first, by allowing a *first boot* time of 90 minutes in which the system does not consume power, as visible in both Fig 7.3a and 7.3b.



(a) Example of successful power budget; the state of charge never dips below 80%.

(b) Example of a non feasible combination of design and mission couple.

7.2 Test mission and results

To begin with, we want to show that the analytical hypothesis, which might be quite abstract and hard to accept out of context, do not prevent the application of the method in concrete scenarios. To this end, we consider a *real mission* for a 6U CubeSat and implement it to prove that the requirements that need to be considered during the preliminary design phase can be represented in this framework.

The second objective is to provide a preliminary validation of the main claim, namely that the solution is unique and optimal for a large number of cost functions. To test this, a brute force approach is used testing a large number of configurations in a neighborhood of the solution to challenge its optimality.

It is important to clarify that our main aim is to validate the algorithm proposed, not the numerical tool or its simulation capabilities. Hence, we will not compare the resulting system with the CubeSat which actually flew the mission. The actual hardware will have been subjected to a detailed design phase in which additional requirements might have been introduced; as we do not have visibility of any requirement besides the high level mission ones, the design will inevitably differ. Furthermore, some assumptions with regard to the technology used will have to be made, as many details of the actual design remain proprietary information.

7.2.1 GOMX-4B

As a reference, we will use GOMX-4B, a spacecraft in a twin CubeSat mission resulting from a collaboration between ESA, GomSpace and the Danish Ministry of Defense. This double mission was chosen for multiple reasons;

- It provides a good representation of the state of the art for CubeSat technology, in terms of both platform and payloads. Development begun in 2015, launching in 2018 and reaching all mission objectives by early 2019.
- As a technological demonstrator mission, there is more information about mission objective than with other commercial missions. Moreover, due to the business model of Gomspace, which both sells the service and the individual components, basic technical information are freely available on their website and can be used to more accurately model the system.
- During a period of internship at Tyvak international, the author has worked as system engineer on the FSS-Cat mission. The mission is very similar to GOMX-4, also using two 6U CubeSats to monitor some propriety of the arctic and even employing some of the same payload, namely Hyperscout. However, as of September 2019, FSScat has not been launched and almost no information on the system is publicly available. Hence, GOMX-4 was chosen as a compromise between a more hands-on familiarity with the mission and the possibility to freely discuss it.
- It features two satellites with somewhat independent missions and objectives but which occasionally collaborate with inter satellite link to test the federated concept.

Mission common goals The two nanosatellites are deployed in a Sun Synchronous Orbit at around 500 km of altitude. They need to perform several experiments and collect data primarily over the arctic, streaming them daily to the primary ground station located in Aalborg (DEN). To demonstrate the federated spacecraft concept, data captured by one satellite is transmitted to the second using the ISL (Inter-Satellite Link) and then finally re-transmitted to ground.

The architecture of both satellites is based on the GomSpace 6U platform; one of the main new subsystems is the NanoPower P60, a modular EPS with independent input and output modules. The space-ground communication capacity of both satellites are improved by the use of High Speed Link in S-band which supports the nominal UHF communication link for uplink and downlink.

GOMX-4A The GomX-4A satellite accommodates the AIS and ADS-B receivers together with a 3 Mpx camera. These instruments operate to track ships and planes over the arctic region. For the ISL operations, the satellite includes two patch antennas in every 2U end and the GomSpace SDR (Software Defined Radio). Dummy masses are added to match the ballistic characteristics of 4B.

GOMX-4B GomX-4B is funded by ESA for In-Orbit Demonstration purposes and carries 5 payloads: the 6U propulsion module from NanoSpace; the innovative ISL (Inter-Satellite Link) from GomSpace; the Chimera board developed by ESA; the HyperScout hyper spectral camera from Cosine and a new Star Tracker from ISIS.

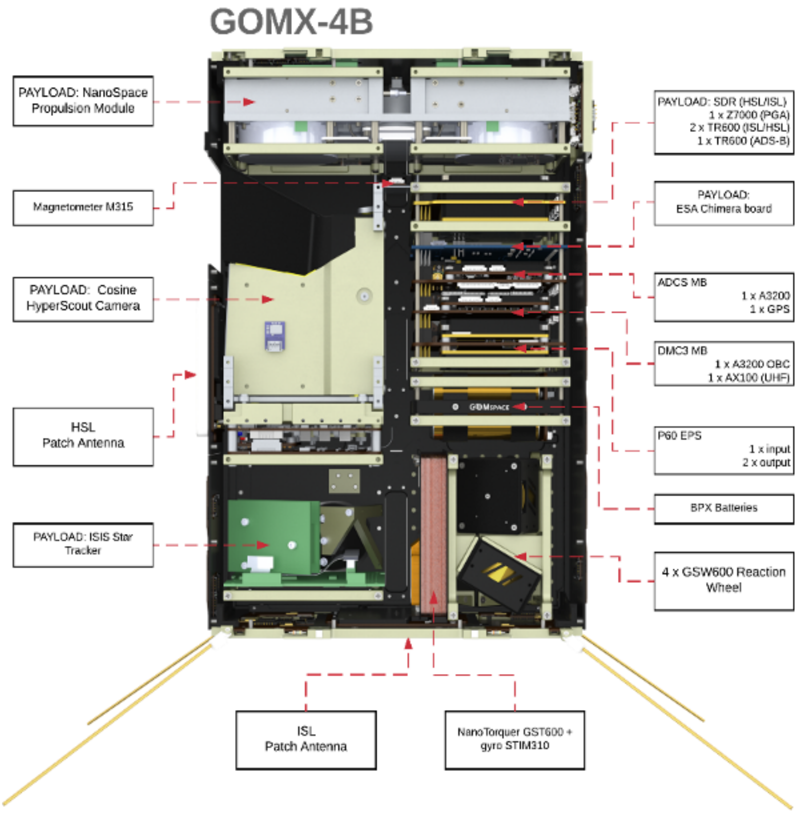

GOMX-4B
PAYLOAD: NanoSpace Propulsion Module
Magnetometer M315
PAYLOAD: Cosine HyperScout Camera
HSL Patch Antenna
PAYLOAD: ISIS Star Tracker
PAYLOAD: SDR (HSL/ISL) 1 x Z7000 (PGA) 2 x TR600 (ISL/HSL) 1 x TR600 (ADS-B)
PAYLOAD: ESA Chimera board
ADCS MB 1 x A3200 1 x GPS
DMC3 MB 1 x A3200 OBC 1 x AX100 (UHF)
P60 EPS 1 x input 2 x output
BPX Batteries
4 x GSW600 Reaction Wheel
ISL Patch Antenna
NanoTorquer GST600 + gyro STIM310
GOMSPACE

Figure 7.4: Internal disposition of the subsystem inside GOMX 4B

We decided to simulate only GOMX-4B; although the two bus are similar, 4B has more payloads to simulate and more complex operations to model. We now review more in detail how we modeled its payloads and bus.

7.2.2 4B-Mission implementation

Given the available information, we will simulate GOMX-4B mission as follows;

- Hyperscout, the hyper-spectral camera, is to be activated above the Arctic region and requires orienting the optical bore sight to nadir. We attributed to the payload a power consumption of 8 W. As can be seen in Fig 7.4, the instrument optical bore sight is approximately perpendicular to a 3U face. To account for the opening, we limit the maximum number of possible solar cells on this face to 4. Attitude for hyperscout observation is depicted in 7.5.

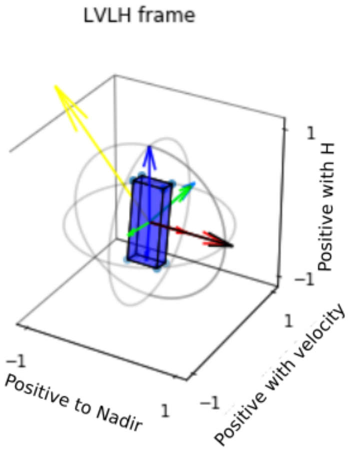


Figure 7.5: Hyperscout pointing Nadir. Reference frame colors: Nadir, direction of orbital velocity, Angular momentum, Sun pointing vector

- Radio Inter Satellite link is scheduled as a window of approximately 8 minutes, with 2 operations per day. For simplicity, we have scheduled operations to be performed over the equatorial region. The bore sight of the antenna needs to be pointed in the direction of the orbital velocity, towards GOMX-4A, as depicted in Fig 7.6. We attributed to the ISL radio a power requirement of 5W.

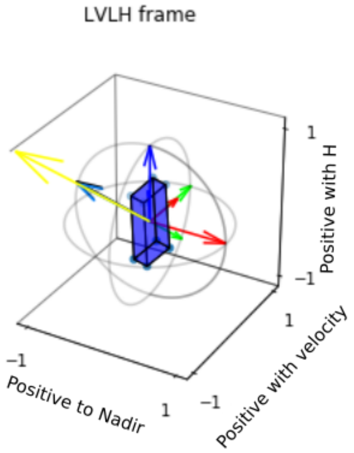


Figure 7.6: ISL patch antenna pointing towards GOMX 4A, ahead on the same orbit. Reference frame colors: Nadir, direction of orbital velocity, Angular momentum, Sun pointing vector

- The Chimera board is to be activated above the South Atlantic Anomaly. As a result, roughly

7 operations per day are performed with an average length of 9 minutes. This experiment does not impose any constraints on the attitude of the satellite. When no attitude constraint is present, the satellite will automatically choose sun pointing, as shown in figure 7.7. We estimated the power consumption of the board at 1.5 W.

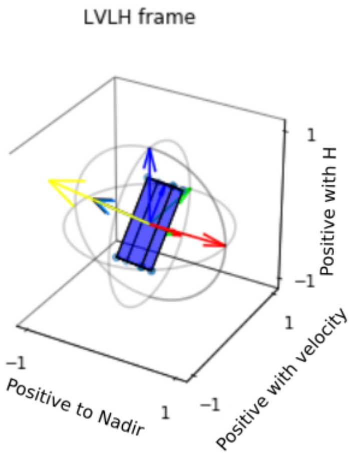


Figure 7.7: When no attitude requirement is present, the default attitude is sun pointing. Reference frame colors: Nadir, direction of orbital velocity, Angular momentum, Sun pointing vector

- To perform the down link, the target ground station is set in Aalborg (Denmark), which results in 5 good passes per day, with an average length of 7.3 minutes. For the radio, we allotted a power consumption of 15 W.

We will not simulate the cold gas motor operations; on the other hand the ISIS star tracker is assumed to be always operational and accounted for as a part of the platform consumption, assessing its power requirement at 1W. Figure 7.8 and table 7.1 summarize the mission parameters.

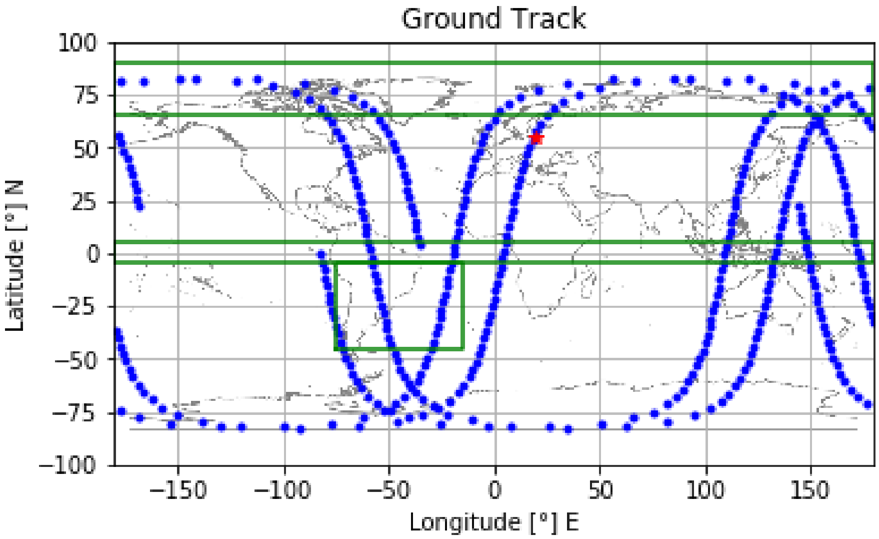


Figure 7.8: Ground track of the first two orbits, Ground Station and active zones for Hyperscout, ISL and Chimera board.

System	Figure	Value
[Eclipse]	avg duration	32.3 min
Radio	GS passes per day	5
Radio	length GS passes	7.3 min
Radio	duty cycle	2.5 %
HyperScout	operations per day	10
HyperScout	operations length	12.2 min
HyperScout	duty cycle	8.8 %
ISL	operations per day	2
ISL	operations length	7.9 min
ISL	duty cycle	1.3 %
Chimera	operations per day	7
Chimera	operations length	8.6 min
Chimera	duty cycle	4.2 %

Table 7.1: High level figures for GOMX 4B mission

Finally, we made some assumptions on basic platform characteristics:

- Baseline bus consumption around 6W (as per website declaration), including attitude determination and on board computer.
- Standard Battery pack 72 Whr, based on 8 separate 18650 cells (as per website)
- Solar cell with efficiency 30 %, effective area of $3 \times 10^{-3}\text{m}^2$

7.2.3 Results

We can test the algorithm proposed in the previous chapters. The LTAN of the sun synchronous orbit is conservatively chosen as 10 AM. The fixed point algorithm converges in less than 10 iterations. It is worth noting that, depending on the effort required in the system design part, each iteration might require considerable amount of time, in the order of tens of seconds.
The chosen configuration is represented by a design point with 2 battery packs, 15 solar cells on a 6U face which is to be pointed towards the sun, and a single reaction wheel per axis. The power budget for this configuration is reported in figures 7.9,7.10,7.11 for different time scales.

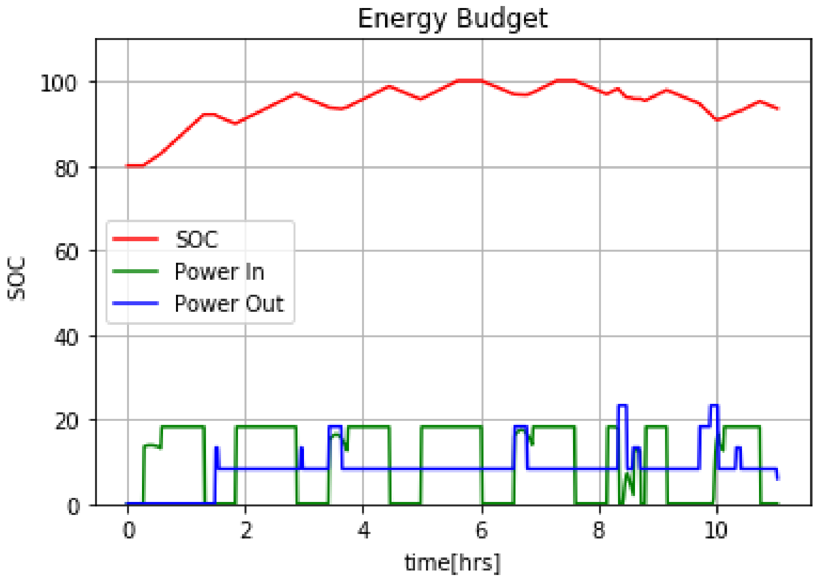


Figure 7.9: State Of Charge (SOC) and power trend (shown with the same scale) for the chosen solution; short duration focus.

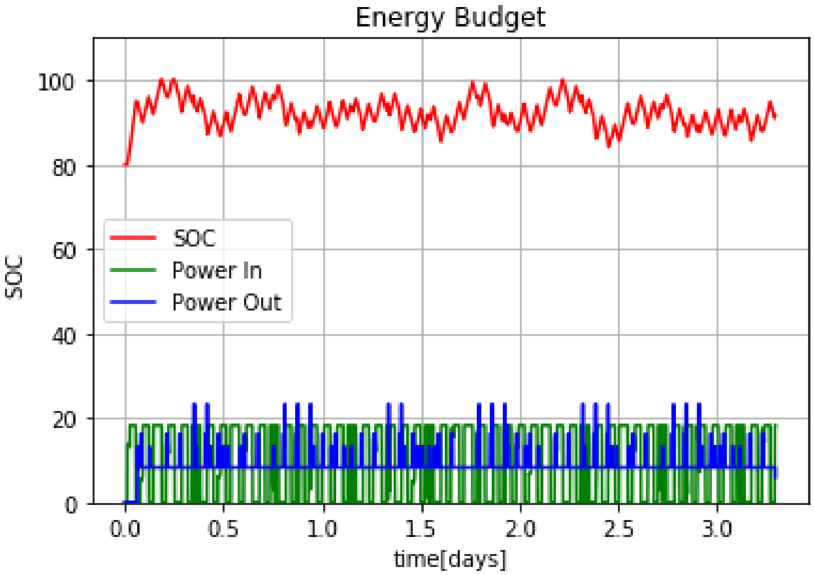


Figure 7.10: State Of Charge (SOC) and power trend (shown with the same scale) for the chosen solution; medium duration focus

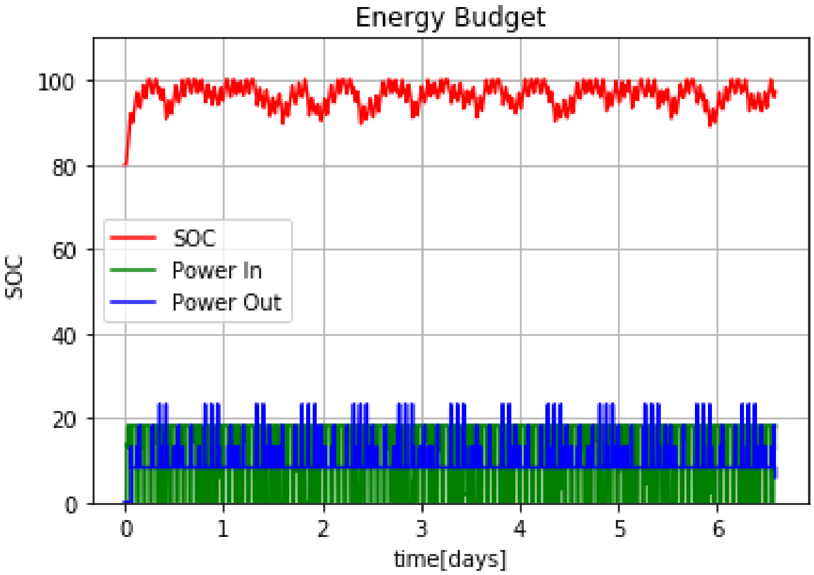


Figure 7.11: State Of Charge (SOC) and power trend (shown with the same scale) for the chosen solution; longer duration focus

We now turn to challenge the optimality of the point suggested by the algorithm. Starting from the proposed best design, we randomly alter the design point by adding or subtracting solar cells on various faces, batteries or RW. For each new configuration we simulate the mission and assess if it is feasible or not; a record is kept of all suitable configurations. Finally we go through the list and check whether any design had requirements which were strictly less demanding than the proposed solution. In our test, we did not find any.

In figure 7.12, we plot a red dot for every feasible solution we find on a 2 projection showing the total number of solar cells vs the total number of battery cells. We can clearly see that no point which uses fewer batteries or fewer solar cells has been found. This reinforces the idea that it is truly a robust optima.

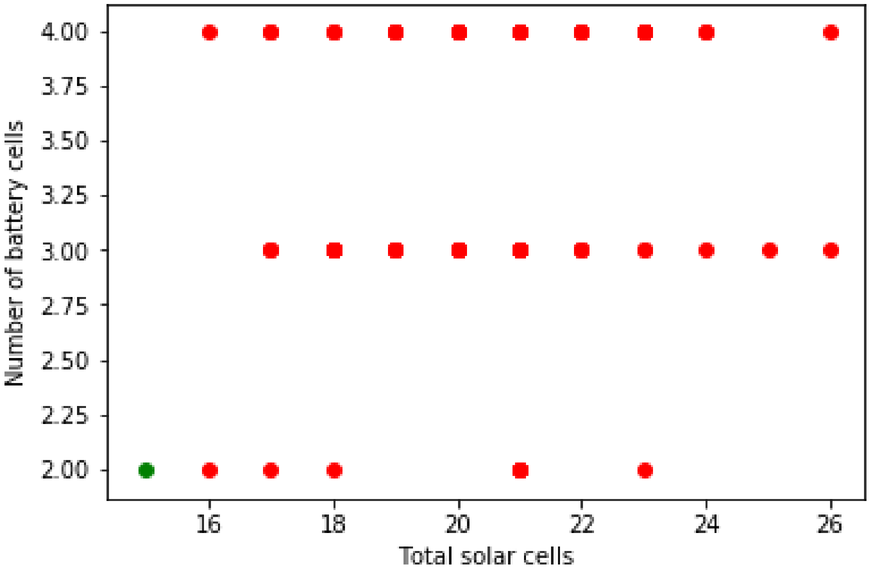


Figure 7.12: Proposed Solution (green) VS other feasible solutions (1000 attempts)

One might rightly wander why some spots in the grid are not filled. In this simplified implementation, the answer is that they simply were not tested. We felt that using a random modification of the optimal design was a good way to remove possible bias on our part; on the other hand, confirming or rejecting a design takes some non negligible amount of time and computational resources, especially for longer missions. Thus a compromise was reached using a thousand attempts.

On the other hand, for a more complex system model, the Pareto front might actually not be a simple straight line. We can imagine that increasing the number of battery packs increases the mass and inertia of the satellite, thus possibly increasing the power requirement. Thus, to remain within the feasibility region, a simultaneous increase in energy storage and production might be required.

Finally, it is important to acknowledge the possibility that a design with few batteries or solar cell does indeed exist, but our random search did not stumble upon it. This could likely be attributed to a bug in the software, as it is not a completely trivial code and testing was limited by time constraints. However, the significant point to make is that this is a numerical validation; it is not a proof. It merely increases the confidence we might have on the fact that it is possible to implement the proposed algorithm and that it appears to work. The proof of its correctness is in the mathematical discussion in the previous chapters.

Refined results To improve the solution, we can break down the battery packs into the elemental cells, and thus increase the resolution with which the algorithm can approximate the optima point. The only difference compared with the previous test is the shrinking of the battery pack to a single battery cell. The optimal design is quite similar to the previous one, but with a reduced total capacity, and thus it is marginally lighter, as reported in table 7.2. Notably, figure 7.13 still proves that the design is a strong optima point.

Figure	Cluster	Monolithic	Unit
Mean Solar Power	9.66	9.66	[W]
Mean Power Out	9.43	9.43	[W]
Max DOD	15.9	13.7	[%]
Satellite Mass	**7.90**	**8.05**	[kg]
Total solar cells	**15**	**15**	[#]
Battery capacity	**117**	**154**	[Whr]

Table 7.2: System comparison betweeen using larger battery packs and cluster of battery cells

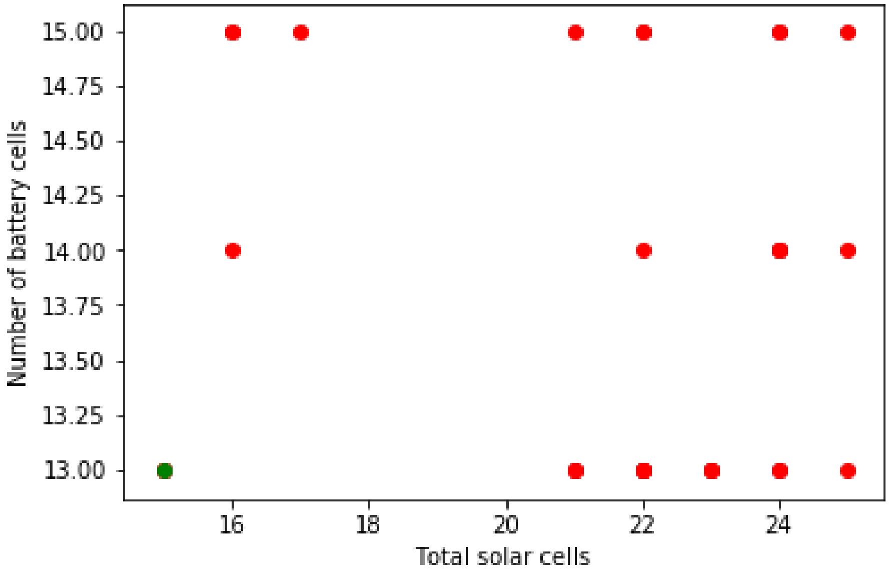


Figure 7.13: Proposed Solution (green) VS other feasible solutions (1000 attempts)

Chapter 8

Conclusions

We started off with the reasons that support multi agent architectures and thus the possibility to extend a cluster beyond the boundaries of the individual system. Considering a generic and dynamic group of actuators, we started chapter 2 with the task of controlling a cluster in an effective way.

Although the control of complex systems is the subject of a great amount of research across several fields, from economics to computer science, only a few solutions were found to be able to improve cluster performances without crippling reliability. Among these, Dual Ascent was selected as a benchmark and a new method was developed to achieve an effective control of a cluster; mathematical proof of convergence are provided in continuous time. Noticeably, the proposed method involved less stringent constraints compared to the traditional Dual Ascent method, which relies on the convexity of the cost function.

To characterize the proposed method in a relevant task, it was applied to the ADCS of a small satellite, which proved to be less straight forward than initially anticipated. To begin with, a hardware prototype was developed to fit the analytical model for the actuator. Several empirical corrections were also proposed to increase the accuracy of the model. Further complications were caused by the significant effect that the angular velocity of the rotor has on the actuator's ability to provide torque and its efficiency in doing so. This coupling confronts any optimization effort with a dynamic problem, for which we offer some possible solutions, but without any formal guarantees. Finally, the proposed method was characterized both with regard to its efficiency compared with standard static allocation and with regard to its stability in the discrete time implementation. In this aspect, we were able to observe that both dual ascent and the proposed method were very robust with respect to noise. However, when increasing the number of agents in the cluster, unexpected behaviors were observed. Specifically, the convergence of the algorithms seemed to no longer be assured as more agents were added. In fact, increasing the number of agents increases the speed at which the cluster state evolves; on the other hand, in a discrete time implementation the size of the integrating time step remains constant, thus eventually becoming too big compared to the speed of the cluster to still approximate the continuous time solution. This phenomena is greatly delayed in the proposed method compared with Dual Ascent.

Expanding the model to actuators featuring multiple inputs and multiple outputs has proven challenging, therefore some additional assumptions are introduced to obtain basic results. To account for multiple outputs we followed the same proof structure used for SISO system, thus proving convergence to stationary point under the constraint, proving the stability of the minima and finally the instability of the maxima and saddle points. The use of an arbitrary cost function is suggested to collapse multiple inputs into a single, *cost-like* value. A considerable amount of work is then poured into finding a less arbitrary strategy, but even under the simplifying assumption of a linear model, only weak results were found. We hypothesized the maximization of the useful life of the system as an additional objective and thus tried to assign relative values to the input resources in agreement with this objective. Thus, having each agent minimize the cost of their actions we could achieve a decentralized maximization of the useful life of the system. Even under a simplifying linear assumption, strong or conclusive results were not reached.

Having the ability to effectively control large clusters, it is possible to either significantly scale up a technology to increase the cumulative throughput, or to increase the granularity of available choices by scaling down the individual component. This is useful to better approximate an optimal design point, which typically is defined on a continuous domain. Exploring this second option, we have reviewed traditional Multidisciplinary Design Optimization techniques. However, using clusters, it is possible to implement a more robust algorithm, which provides stronger theoretical assurances regarding the optimum. A long and rather dry discussion of technical details is presented in chapter 5, offering mathematical proofs for the main claim. A long section is devoted to the justification of the hypothesis of the model, namely to present sufficient conditions for the design process to be a contraction map. An initial simplified scheme is presented and subsequently expanded into more realistic and complex models.

In an attempt to render the results of the previous chapter directly useful and to set the stage for the final chapter, a short exposition is devoted to the tailoring of the method to a cubesat mission. The main subsystems, such as the ADCS, the power system, the thermal control system etc are considered individually and cast into the nomenclature of the model presented. Only analytical considerations are taken into account.

In the final chapter we offer a preliminary validation of the algorithm with a computational implementation in Python. The framework is designed with CubeSats for earth observation missions in mind. The most essential capability is to be able to simulate a generic mission and the behavior of the specified satellite with the intent of assessing mission feasibility. On top of this framework, the proposed algorithm is implemented and tested on a recent mission, GOMX 4B. The optimality of the proposed solution is then challenged by randomly perturbing the design point and assessing the modified system capability to achieve the mission objective. In the test performed, the algorithm solution was confirmed as optimal. It is to be noted however, that the numerical validation does not constitute a proof, it only increases our confidence in the possibility to fruitfully apply the method to real missions; proof of the optimality claim have been addressed with the analytical model.

Future work

Multi agent operations with CubeSats are a very exciting topic for research and industry alike. In the coming years, if not months, both NASA and ESA will launch their demonstrators (called C-Pod and RACE respectively) to test in orbit assembly. The hardware for miniaturized and autonomous docking seems to be almost ready to enable very ambitious in-space assembly projects. On the other hand, large and mega constellations, such as the Space X Starlink, have already begun deployment so it is not yet clear whether the space industry will choose the first declination of multi agent system, the second or both.

Depending on the interest in large in-space assemblies, this work might become a mere curiosity or a useful starting point. Although even constellations may benefit from some decentralized coordination, physical docking enables a much simpler and more reliable path to share a broader range of resources. To further develop in space assembly, a more formal characterization of convergence under discrete time implementation would be useful. The most important hurdle to overcome however would be the development of a proper and complete theory for multiple input agents, although the author does not have any good suggestion on how this could be achieved.

The second part of this work, devoted to a new tool for system design, has revealed to be quite technical and, in its current form, it is highly unlikely to be considered by the industry. In order to become useful, a proper numerical implementation would need to be developed to automate most, if not all, of the most tedious details and to provide reliable off the shelves model at least for standard subsystems. At the current stage however, the algorithm capabilities do not seem to be enough to motivate the investment of the considerable amount of resources that would be required to produce a commercial product.

Nonetheless, it is the author's belief that the algorithm presented is significantly different from the standard MDO approach to warrant further examination. Although not as general as other methods, it is able to provide interesting assurances which would otherwise be prohibitively expensive to obtain numerically.

List of Figures

2.1 Information paths in Centralized (A) and Decentralized (B) algorithms. In the centralized approach, the node C has to process all information about the network. With the distributed approach, each agent handles only a subset of the total information handling. . . . 13

2.2 Single input-Single output model for a generic agent . . . 17

2.3 Example of efficency curve (smooth line) and cost curve/resource consumption curve (dashed) . . . 17

2.4 A cluster of *n agents* with parallel consumption and production. . . . 17

3.1 The standard electro-mechanical model for a DC engine . . . 31

3.2 3D rapresentation of Power necessary for a given torque at ω . . . 33

3.3 A visualization of P as a function of T and ω. . . . 33

3.4 Schematic representation of the test RW model . . . 34

3.5 Picture of the hardware set up; brushless motor under the inertia disk, ESC (in red, connected to the motor),RPM sensor (connected to the motor by yellow, red and black wires), Arduino/controller (in green), Amp sensor (in blue) and power supply unit. . . . 35

3.6 Time dependent RPM readings for different constant input signals (in PWM); these tests confirm that a constant PWM input leads to a constant RPM output, after a transitory period (not shown in figure). . . . 36

3.7 Time dependent mAmp readings for various constant input signals (in PWM); current reading show large deviation with respect to the mean, which appears to be in steady state. . . . 37

3.8 Control input (PWM [%]) Vs average steady state output (RPM) . . . 37

3.9 Steady state power consumption map against average data points . . . 37

3.10 RPM vs Time signals for haviside step input of different amplitude (in PWM); the transitory response is shown and suggests that the motor dynamics can be approximately consdiered linear. . . . 39

3.11 Fit for control stiffness parameter k . . . 41

3.12 A set of output signals for various RPM/S requests, both increasing and decreasing 42

3.13 Request vs output; measured angular acceleration is averaged over a signal of approximately 20 seconds . . . 42

3.14 Linear fit for dynamic current as a function of requested angular acceleration . . . 45

3.15 Depiction of the pilot torque signal, parameteric with τ and α . . . 47

3.16 Using different values of α the optimal allocation among two RW acting in the same direction to supply $\approx$ 0.12 mN m changes. The green line shows all the combinations of T1 and T2 able to meet the requested output torque. The red and bluecurved lines show equi-cost lines for two different values of α. As we would expect, the optimal regions, circled in red and blue change as well, depending on the choice of α / cost model. (Graphically we can recognize the constrained stationary points as the points in which the equi-cost lines are tangent to the constraint line) 48
3.17 Cost functions according to Eq. 3.22 for the Analytical and Empirical model. Two different values of α are considered ($\tau = 100$ms). 49
3.18 Different probability density function. 51
3.19 Large scale picture of expected power consumption $\mathcal{P}$ using different pdf. 52
3.20 Small scale picture of expected power consumption $\mathcal{P}$ using different pdf. 52
3.21 A more useful visualization of ε^* as a function of P_{el} and ω 54
3.22 Efficency at different ω 54
3.23 Large torque request (small sat) expected power for n=1,2,5. 55
3.24 Detail of expected power for n=1,2,5 55
3.25 Different $\mathcal{P}$ using static division or a central allocator. 57
3.26 Percent savings in expected power consumption 57
3.27 Comparison between optimal allocation vs distributed allocation, estimated power requirement with constant pdf 57
3.28 Comparison between optimal allocation vs distributed allocation, power savings in % with constant pdf 57
3.29 Results with the AA model. Parameter for the test in table 3.9 58
3.30 Results with the A0 model. Parameter for the test in table 3.9 59
3.31 Results with the Modified model. Parameter for the test in table 3.9 59
3.32 A set of tests with a single agent, multiple initial conditions. On the top, convergence to the requested torque, in red. On the bottom, the efficiency plot. 61
3.33 Average number of steps to convergence and standard deviation (at 1σ) for the proposed method (in blue) and Dual Ascent (in green). The red dots mark DA batches in which at least one test has failed to converge in the allotted time. 62
3.34 Each agent in Dual Ascent is started from different value of λ_0.The proposed method is identical to the ideal case and reported for reference. 63
3.35 Single component, noise as per table3.11, 5 different runs. Profiles are more jagged due to the noise contribution. 64
3.36 20 Agents, with measurement noise, identical λ_0. 65
3.37 A cluster of 3 RW following a target sinusoidal signal at approximately 10 Hz. The units of the x axis are timesteps, each 10 ms long. 66

4.1 Two possible target in the resource space 83
4.2 Assuming both option A and B feasible, A costs a lot less than option B since it has a smaller projection on the cost vector $\vec{c}$. Note that the arrow represent the movement in the resource space, which is $-\mathbf{E}\phi$; the sign is negative because resources are being consumed. 88
4.3 Vector field of cost function (in blue) and cost minimization strategy (in green). . . . 89

5.1 An abstraction model of the design process. A set of system parameter are assumed and tested against mission objective. If the design results to be feasible, the initial hypothesis of system parameters are checked against subsystem technological capabilities. 96
5.2 A possible hirachy of system design; each level confirms or refutes the hypothesis made at the level above. 96
5.3 Condition **A,B** identify sets $\mathcal{A}, \mathcal{B}$, while $\mathcal{C}$ is a scalar function defined on $\mathcal{B}$, and minimized at the shown point. 98
5.4 A diagram of the conditions to check for consistency at the subsystem level. 100
5.5 An example of redefinition of the technological curve, in red dashed. 106
5.6 Net energy flow; positive sign indicates energy leaving the battery 117

7.1 The basic iterative process; we obtain requirements based on mission description and some assumptions, we then design each subsystem and finally update initial assumptions. 145
7.2 An example of mission definition, with ground track in blue, selected ground stations in red and areas of interest in green. 147
7.4 Internal disposition of the subsystem inside GOMX 4B 151
7.5 Hyperscout pointing Nadir. Reference frame colors: Nadir, direction of orbital velocity, Angular momentum, Sun pointing vector 152
7.6 ISL patch antenna pointing towards GOMX 4A, ahead on the same orbit. Reference frame colors: Nadir, direction of orbital velocity, Angular momentum, Sun pointing vector 152
7.7 When no attitude requirement is present, the default attitude is sun pointing. Reference frame colors: Nadir, direction of orbital velocity, Angular momentum, Sun pointing vector 153
7.8 Ground track of the first two orbits, Ground Station and active zones for Hyperscout, ISL and Chimera board. 154
7.9 State Of Charge (SOC) and power trend (shown with the same scale) for the chosen solution; short duration focus. 155
7.10 State Of Charge (SOC) and power trend (shown with the same scale) for the chosen solution; medium duration focus 156
7.11 State Of Charge (SOC) and power trend (shown with the same scale) for the chosen solution; longer duration focus 156
7.12 Proposed Solution (green) VS other feasible solutions (1000 attempts) 157
7.13 Proposed Solution (green) VS other feasible solutions (1000 attempts) 158

List of Tables

3.1 Reference values for a RW 32
3.2 The summary of the outcome of a set of tests 36
3.3 Best fits for the parameter k, with 95% upper and lower bounds 40
3.4 Measures from closed loop actuation 43
3.5 Prediction current of the two model 45
3.6 Coefficients obtained by fitting the empirical data for various units of inputs and outputs. 46
3.7 Response times of each of the phases in the control loop. 46
3.8 Coefficients for the expected power consumption based on analytical and empirical models. 51
3.9 Coefficient for the various efficiency study 60
3.10 Values used in the simulation with the A0 model for power (and cost). 61
3.11 Coefficient for the noise functions 64

7.1 High level figures for GOMX 4B mission 154
7.2 System comparison betweeen using larger battery packs and cluster of battery cells . 158