UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Model-Based Control Techniques for Automotive Applications

Ph.D. candidate
Fabio Maran

Advisor
prof. Alessandro Beghi

Director
prof. Matteo Bertocco

Coordinator
prof. Carlo Ferrari

Ph.D. School in
Information Engineering

Department of Information Engineering
University of Padova
2013

Summary

Two different topics are covered in the thesis.

Model Predictive Control applied to the Motion Cueing Problem

In the last years the interest about dynamic driving simulators is increasing and new commercial solutions are arising. Driving simulators play an important role in the development of new vehicles and advanced driver assistance devices: in fact, on the one hand, having a human driver on a driving simulator allows automotive manufacturers to bridge the gap between virtual prototyping and on-road testing during the vehicle development phase; on the other hand, novel driver assistance systems (such as advanced accident avoidance systems) can be safely tested by having the driver operating the vehicle in a virtual, highly realistic environment, while being exposed to hazardous situations. In both applications, it is crucial to faithfully reproduce in the simulator the driver's perception of forces acting on the vehicle and its acceleration. This has to be achieved while keeping the platform within its limited operation space. Such strategies go under the name of Motion Cueing Algorithms.

In this work, a particular implementation of a Motion Cueing algorithm is described, that is based on Model Predictive Control technique. A distinctive feature of such approach is that it exploits a detailed model of the human vestibular system, and consequently differs from standard Motion Cueing strategies based on Washout Filters: such feature allows for better implementation of tilt coordination and more efficient handling of the platform limits.

The algorithm has been evaluated in practice on a small-size, innovative platform, by performing tests with professional drivers. Results show that the MPC-based motion cueing algorithm allows to effectively handle the platform working area, to limit the presence of those platform movements that are typically associated with driver motion sickness, and to devise simple and intuitive tuning procedures.

Moreover, the availability of an effective virtual driver allows the development of effective predictive strategies, and first simulation results are reported in the thesis.

Control Techniques for a Hybrid Sport Motorcycle

Reduction of the environmental impact of transportation systems is a world wide priority. Hybrid propulsion vehicles have proved to have a strong potential to this regard, and different four-wheels solutions have spread out in the market. Differently from cars, and even if they are considered the ideal solution for urban mobility, motorbikes and mopeds have not seen a wide application of hybrid propulsion yet, mostly due to the more strict constraints on available space and driving feeling.

In the thesis, the problem of providing a commercial 125cc motorbike with a hybrid propulsion system is considered, by adding an electric engine to its standard internal combustion engine. The aim for the prototype is to use the electrical machine (directly keyed on the drive shaft) to obtain a torque boost during accelerations, improving and regularizing the supplied power while reducing the emissions.

Two different control algorithms are proposed

1. the first is based on a standard heuristic with adaptive features, simpler to implement on the ECU for the prototype;

2. the second is a torque-split optimal-control strategy, managing the different contributions from the two engines.

A crucial point is the implementation of a SIMULINK virtual environment, realized starting from a commercial tool, VI-BIKEREALTIME, to test the algorithms. The hybrid engine model has been implemented in the tool from scratch, as well as a simple battery model, derived directly from data-sheet characteristics by using polynomial interpolation. The simulation system is completed by a virtual rider and a tool for build test circuits.

Results of the simulations on a realistic track are included, to evaluate the different performance of the two strategies in a closed loop environment (thanks to the virtual rider). The results from on-track tests of the real prototype, using the first control strategy, are reported too.

Sommario

Nella tesi vengono trattati due argomenti distinti.

Model Predictive Control applicato al Motion Cueing Problem

Gli ultimi anni hanno visto un'interesse sempre crescente nei confronti dei simulatori di guida dinamici, con lo sviluppo e la diffusione nel mercato di nuovi soluzioni. I simulatori di guida giocano infatti un ruolo fondamentale nello sviluppo di nuovi veicoli e dei vari dispositivi di aiuto alla guida: infatti, da un lato la presenza di un guidatore in un simulatore permette ai produttori in ambito automotive di colmare il divario fra la prototipazione virtuale e i test su strada nella fase di sviluppo del veicolo; dall'altro, i nuovi sistemi di assistenza alla guida (come ad esempio le procedure di advanced accident avoidance attualmente in fase di sviluppo) posso essere testati in totale sicurezza ponendo il pilota in un contesto virtuale altamente realistico, simulando le situazioni di pericolo. In entrambe queste applicazioni risulta cruciale riprodurre fedelmente nella piattaforma la percezione che l'essere umano avrebbe, all'interno del veicolo reale, delle forze agenti sul mezzo e le conseguenti accelerazioni. Questo task deve essere compiuto tenendo conto dei vincoli fisici del simulatore, all'interno dei quali deve avvenire il moto. Le strategie utilizzate per perseguire questo obbiettivo vanno sotto il nome di Motion Cueing Algorithms.

Il presente lavoro intende illustrare una particolare implementazione di un Motion Cueing Algorithm, basato sulla tecnica di controllo nota come Model Predictive Control. Una delle principali caratteristiche di questo approccio è lo sfruttamento di un modello del sistema vestibolare umano, e questo (assieme ad altre features) lo rende differente dalle strategie standard di Motion Cueing: esso permette infatti una migliore realizzazione della tilt coordination e una più efficiente gestione dei limiti di piattaforma.

L'algoritmo è stato testato sperimentalmente su una piattaforma innovativa, dalle dimensioni ridotte, con l'aiuto di piloti professionisti. I risultati dimostrano come l'algoritmo basato su MPC permetta di gestire efficientemente l'area di lavoro del simulatore, limitando la presenza di tutti quei comportamenti tipicamenti associati alla motion sick-

ness, garantendo nel contempo un approccio molto più semplice e concreto alle procedure di tuning, rispetto alle procedure classiche.

In più, la disponibilità di un efficace driver virtuale permette lo sviluppo di strategie predittive affidabili: nella tesi sono riportati alcuni iniziali risultati simulativi in tal senso.

Tecniche di Controllo per un Motociclo Ibrido Sportivo

La riduzione dell'impatto ambientale dei sistemi di trasporto si sta affermando come una priorità sentita a livello mondiale. I veicoli a propulsione ibrida hanno dimostrato avere un grande potenziale a questo riguardo, e svariate soluzioni sono ormai diffuse sul mercato per quanto riguarda i veicoli a quattro ruote. A differenza delle automobili, e pur essendo considerati la soluzione ideale per la mobilità urbana, l'applicazione della propulsione ibrida a motociclette e scooter non ha ancora avuto una diffusione significativa, e ciò è dovuto in gran parte ai grossi vincoli di spazio e all'impatto della propulsione additiva sul feeling alla guida.

In questa parte della tesi viene considerato il problema dell'applicazione della propulsione ibrida a una motocicletta 125cc in commercio, aggiungendo una macchina elettrica al motore termico presente di serie. Lo scopo, per il particolare prototipo, è sfruttare la macchina elettrica (installata in modo solidale all'albero motore) per fornire un incremento alla coppia erogata durante le accelerazioni, migliorando e regolarizzando la potenza della moto e riducendo nel contempo le emissioni di gas nocivi.

Due algoritmi di controllo sono proposti per la gestione del motore elettrico e degli accumulatori

1. il primo è basato su una euristica standard con caratteristiche adattative, più semplice da implementare nella ECU per la prototipazione;

2. il secondo è basato su una strategia di controllo ottimo con lo scopo di gestire in maniera ottimale la coppia erogata da entrambi i motori.

Elemento cruciale è l'implementazione di un ambiente virtuale SIMULINK realizzato integrando un tool in commercio, VI-BireRealTime, per la valutazione degli algoritmi. Il modello del motore ibrido è stato implementato ex-novo, e così anche un (semplice) modello di batteria, derivato con interpolazione polinomiale dalle caratteristiche riportate nei data-sheet. Il sistema di simulazione è completato dalla presenza di un virtual rider e di un tool per la realizzazione di circuiti di test.

Sono riportati i risultati delle simulazioni su un tracciato realistico per valutare le differenti performance delle due strategie in catena chiusa (grazie al rider virtuale). Sono riportati anche i risultalti su pista del prototipo realizzato.

Contents

Part I

Model Predictive Control applied to the Motion Cueing Problem

1

Introduction

In the recent years, *dynamic* driving simulators have been playing an important role in the automotive world. The first applications were in the racing context, focused on driver's training and virtual vehicle set-up. Nowadays, applications of such systems are becoming more and more numerous and diverse, involving field as security control systems (e.g., accident avoidance), medical rehabilitation, virtual prototyping. Automotive OEMs exploit driving simulators to cut down the costs for prototyping, by anticipating the on road vehicle behavior. Furthermore, such systems allow to ease the development process of the various vehicle components, by testing different hardware and software solutions, by resorting to sophisticated Hardware-In-the-Loop (HIL) tools, in a safe and realistic virtual environment. In a different perspective, realistic dynamic simulators are crucial to develop detailed driver behavior models to devise accident avoidance strategies, for example by putting the driver in simulated dangerous condition and measuring his/her reactions. Also, assessment of driver performance under stress conditions (i.e., adverse weather

conditions, endurance driving, etc.) can be more effectively performed by using dynamic simulators with a high degree of immersion into virtual environments. To this regard, ever more effective Hazard Perception Test can be devised and used as requirements for achieving a driver license. In this scenario, the effectiveness of a dynamic platform is deeply related to the capability of reproducing in the most realistic way the feelings that the driver would have inside a real vehicle: this is the role of the *Motion Cueing* (MC) strategy.

Motion Cueing is the name of the algorithms designed for transforming vehicle accelerations into motion commands to the platform, aiming at accurately replicating the forces acting on the driver and guaranteeing that the machine stays inside its physical limits. Clearly, this is a very complicated part of a dynamic simulator, due to the complex nature of the human perception system, which involves different organs and senses. In fact, it is not clear yet, from a physiological point of view, the role and priorities of stimula of different nature to the overall perception of accelerations and force. It is however well established that a coordinated visual-motion action is fundamental for achieving satisfactory performance of a MC algorithm. Given the above motivations the necessity of a perceptive model within the MC algorithm becomes evident.

Beside the improvements realized during the years, in most dynamic simulators the Motion Cueing Algorithms are based on a sequence of passive filters Nahon and Reid (1990) combined together. The standard procedure is implemented as to

- remove low frequency components of accelerations and velocities obtained from the vehicle dynamic model. In this way, only fast variations are kept, which are easier to be reproduced in a limited space environment;
- transfer part of the low frequency translational accelerations to the angular dynamic using a low pass filter (*tilt coordination*). This is a common way of tricking the human perception, with the help of visual cues, to exploit the gravity force to reproduce at least a part of the low frequency accelerations;
- limit the platform motions with a further high pass filter to keep the platform in a neutral position, i.e. eliminating low frequency components that could lead to infeasible positions for the platform actuators. This is commonly called *Washout Action*.

This simple strategy has seen a wide range of implementations over the years (Conrad and Schmidt, 1971; Nahon and Reid, 1990). However, it has some shortcomings:

- being a filtering based approach, it is not possible to guarantee stimuli consistency between the dynamic simulation environment and the real platform movements,

i.e. it is hard to find a good compromise for complex vehicle maneuvers (e.g. a complete track lap in racing applications);

- it cannot explicitly handle hard constraints on the platform movements and accelerations (both regarding actuators limit and perceptive reliability);
- it is not possible to exploit any available information on the driver's behavior in the future;
- the tuning of the algorithm is in general difficult, since it is not easy to give physical interpretation to most of the parameters (principally filter gains and cut-off frequencies).

These elements characterize the most commonly used procedures. Over the years, there have been modifications and improvements, with the introduction of adaptive (Parrish, Dieudonne, Bowles, and Martin, 1975) and optimal control (Sivan, Ish-Shalom, and Huang, 1982) features, but without significantly modifying the main algorithm.

Recently, a novel approach to motion cueing has been proposed by Dagdelen, Reymond, Kemeny, Bordier, and Maïzi (2009) and Augusto and Loureiro (2009), based on a strategy already consolidated in the field of industrial process control, namely, Model Predictive Control (MPC). MPC is a model-based, *constrained*, *optimal* control methodology that allows to effectively handle limits on the working space and to exploit information on future reference signal. In the idea developed in those works the motion cueing strategy integrates a model of the human perception systems and takes advantage of predictions of the future trajectory to fully exploit the platform working area. In these early works, the proposed solutions are not suitable for experimental application in real situations; as they don't take into account many of the aspects that characterize an effective and reliable MC system (e.g., real time constraints, tuning procedures and so on). Moreover, they focus on investigating the prediction capabilities rather than on taking advantage of the optimization approach of MPC.

The MC algorithm described in this part of the thesis has to be considered one of the first real time implementation of an MPC-based Motion Cueing strategy, developed from scratch and eventually run on an experimental environment. The idea of the algorithm comes from the previously cited works, but with a deeply different development and rationale, with each step realized with the final aim of obtaining a procedure ready to be applied in the real world. One of the key elements is the development of a reliable vestibular model. Being a model-based approach, the availability of such information about the dynamics of the human perception allows a great improvement on the reliability of the signals that will be fed to the physical device, in terms of realistic motions. As

already mentioned, in the work by Dagdelen et al. (2009) a first attempt of exploiting such information has been presented, but the simplicity of the illustrated model and the greater emphasis on the prediction phase reported in the paper seems not not provide effective results. Moreover, other MPC approaches as the one by Augusto and Loureiro (2009) seems to make use of reference signals that might result in undesired behaviours. In the present study, a major review of the literature has been carried on, starting from aerospace applications, to derive more detailed models which have been refined with the help of practical tests by professional drivers and medical consultants. The overall modelization is completed by the integration of a (simple) mechanical model of the platform.

The prediction phase is quite a difficult task for this kind of application, involving a human being in the loop (whose behaviour could be hard to anticipate): in this sense, the main idea is to exploit most the *optimal control* aspect of the MPC methodology, together with the natural integration of the *constraints*. The considered cost function, although being quadratic to reduce the complexity of the problem, takes into account the tracking error, the control signal and its derivative (Wang, 2009), aiming at maximizing the performance while keeping the control signals as regular as possible. Constraints are another fundamental feature of the MPC approach: they not only allow to take into account in a rational way the constructive limits of the device, but also to avoid undesired physical behaviours (e.g. different signs of the accelerations with respect to the visual cues) that may lead to sickness and that are very hard lo manage with the traditional methods.

The effectiveness of a MC algorithm is mostly determined by its *tuning* phase. The proposed strategy introduces a newel approach to this crucial step: both the weights in the cost functions and the constraint values become *tunable parameters*. In this way, the parameters that are regulated have an immediate physical meaning, with a double advantage

1. making the learning phase faster for the operator, since everything is much more immediate to understand and intuitive to manipulate;

2. improving the feedback from the driver, whose suggestions can be immediately applied.

In this sense, it will be shown how the length of the prediction window (i.e. the number N_P of future steps taken into account for tracking) becomes an important parameter to be tuned. As common practice in applied MPC, the reference signal is considered to be constant over the prediction window, because of the difficulty of obtaining a reliable

future signal and, most of all, the increase in terms of resolution complexity.

The resolution of the constrained optimization problem is the hardest part of any MPC application, in particular in the presence of strong *real-time* request, as in the case of MC where fast dynamics are involved. Among the different possibilities given in literature (Bemporad, Morari, Due, and Pistikopoulos, 2002; Boyd and Vandenberghe, 2004; Wang and Boyd, 2010), the chosen approach is a particular implementation of the Active Set strategy, qpOASES Ferreau, Bock, and Diehl (2008), which has a number of features well suited for the considered problem.

As a final validation of the research, the algorithm is evaluated on the field by professional test drivers using an innovative dynamic simulator, VI-DRIVESIM (VI-Grade, 2012), which has the advantage of being a small, low cost platform with a high number of Degrees of Freedom (six, almost decoupled)

Some remarks on a possible implementation on FPGA are also given as topics for future research.

The first part of the thesis is organized as follows.

In **Chapter 2** the general Motion Cueing problem is described, with an overview of the most common driving simulator architectures, and a detailed explanation of the classic approach and its problems and limitations.

In **Chapter 3** the proposed Model Predictive Control procedure is illustrated in details. After an introduction on the general MPC framework, the derivation of the Vestibular Model will be presented, and the overall model will be shown. The optimization strategy will be examined as well, and in the end all the features that characterize the tuning procedure and the rationale behind this methodology will be detailed.

In **Chapter 4** the test platform will be presented and the results from simulations and practical tests analysed and compared. Simulation results that take advantage of prediction will be considered too, even if not tested on the real platform. More, a possible hardware implementation will be introduced, based on Field Programmable Gate Arrays (FPGA) devices.

In **Chapter 6** concluding remarks will be given.

2

The Motion Cueing Problem

The aim of dynamic simulators is to provide the driver with motion feedbacks, called *Motion Cues*, and thus to increase the realism of the simulation, whatever the purposes of the simulation are. This is the goal of the Motion Cueing (MC) algorithm, the procedure that defines the movements of the mechanical system based on the current physical information (acceleration and angular rate) given by the software-simulated vehicle. The geometrical structure of the motion system and its related motion envelope is a major factor for the Motion Cueing, and defines both the ability to present certain motion cues as well as their limitations. Motion Cueing has been implemented over the years following the same main idea, and despite some modifications and integration of new features the procedure has remained substantially unchanged. The development of new, high-performance simulators, as well as their adoption for virtual prototyping, training, rehabilitation and other purposes (Straus, 2005; Slob, 2008) call for new solutions that overcome the limitations of the classic approach. In particular, *simulator sickness* is a

critical drawback in simulators, that may strongly limit their adoption (Straus, 2005).

In this chapter, the classic approach to MC will be presented. First an overview on simulators history and the different solutions adopted over the years will be given; then the main idea behind the classical approach will be explained, analyzing the limitations and problem still affecting these procedures.

2.1 Driving simulators

The idea of motion simulation find its origins at the beginning of the 20th century, and the flight simulator by the French flight training school ANTOINETTE can be considered the first rough example of dynamic simulator: it comprised a half-barrel mounted on a universal joint, with flight controls, pulleys, and stub-wings to allow the pilot to maintain balance while instructors applied external forces (Vincenzi, Wise, Mouloua, and Hancock (2008), figure 2.1). The development of driving simulators in the early years of the century aimed

Figure 2.1: Antoinette Trainer, the first dynamic simulator.

to assess the skills and competence of public transit operators. Over the next four decades, mockup automobiles were equipped with devices to test drivers' responses to various stimuli and in some cases, mechanical moving scenes or filmed road scenes were shown. By the 1960s, a number of automobile manufacturers, automobile insurance companies, military agencies, universities, and aerospace companies used film approach simulators for studies involving a variety of visual displays. A major improvement in this period was the introduction by Stewart (1965) of the parallel, six Degrees of Freedom (DOFs)

system well known from then on as *Stewart platform*, reported in figure 2.2. This consists

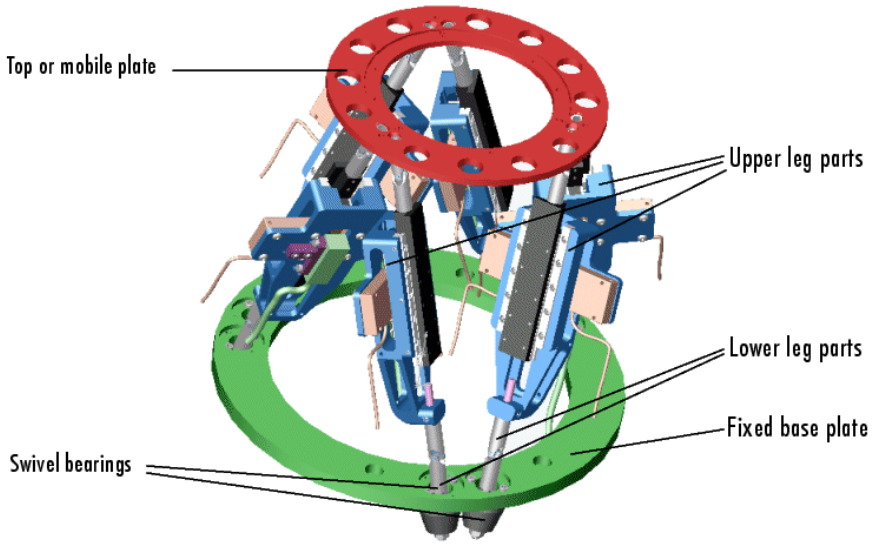


Figure 2.2: The Stewart platform.

of a platform, one face of an octahedron. The base, the opposing face, connects to the platforms by six struts of the octahedron. These struts allow for positioning the platform in the six DOFs: platform orientation and position vary. This device quickly became the core of the most part of driving simulators: the usual structure provides for an actual motor vehicle linked to a computer for data collection and analyses, while various road images are projected unto large screens within a dome-like structure. The evolution over the decades saw the replacement of hydraulic actuation with electric servo technology.

Increasing their spread, driving simulators have found various areas of use, from entertainment to research and advanced training. Following the classification by Slob (2008), they are subdivided on the base of a "fidelity" criterion as follows

1. *low-level* simulator: the driver sits in a car seat, preferably inside a car, which is fixed to the ground. This is called a fixed base (FB) simulator. The driver looks to a screen, which is fixed too (FS). The screen is designed such that the view angle is as large possible and this can be done using multiple screens, or (preferably) a large convex screen. Steering wheels and pedals are often equipped with force feedback and sound system takes care of the audio feedback.

2. *mid-level* simulator: it consists of a car accelerated in one degree of freedom. The screen can be fixed (FS) or can move along with the car. Force feedback is applied to the steering wheel and a sound system takes care of the audio feedback. The actuated DOF at a 1-DOF simulator is often a y-sled, x-sled or a yaw-table.

3. *high-level* simulator: it actuates the payload in at least six DOFs. The payload might be accelerated in additional DOFs (introducing redundant DOFs), e.g. to allow longer planar excursions. The payload often consists of a dome with a car (or its interior) inside of it and graphics projected such that a view angle of at least 220 degrees is covered. The largest driving simulators consist of dome on a turntable, mounted on a hexapod fixed to an xy-table.

Many works give insights of implemented simulator solutions (e.g. see reports by Straus (2005) and Slob (2008), and paper by Wang, Zhang, Wu, and Guo (2007)): the overview by Slob (2008) is adequate for the aim of this dissertation, focusing on high-level simulators.

Volkswagen The first high-level driving simulator was built by Volkswagen in the early '70s and consisted of a car on a 3-DOFs motion system. The motions were driven by a turntable (yaw) and a roll and pitch mechanism. A single, flat screen was mounted in front of the driver sitting on its seat at a platform, and no further car interior was reproduced at the platform.

VTI The Swedish Road and Traffic Research Institute VTI is a long time active player in the world of driving simulation, proposing y-sled in their designs. The first product was (in 1984) the VTI-I, a four DOFs simulator. A half car with a screen fixed in front of it on a motion platform was accelerated in roll, pitch and yaw, on a y-sled (sway). At the end of the 1980s, VTI renewed the VTI-I to the VTI-II to implement a truck simulator, which has a higher payload than the passenger car simulator. In 2004 this design was again upgraded to the VTI-III, increased in size and equipped with a vibration table, allowing high frequent road rumbles to be experienced by the driver.

Daimler-Benz The initial "Daimler-Benz" Driving Simulator was introduced in 1985 and was the first driving simulator to be driven in six DOFs. An hydraulic hexapod, which was a special design for this simulator, realized the largest motion envelope at that time. A car or truck cabin was situated inside a dome on which 6 CRT projectors displayed a 180 degrees field of view. In 1993 the simulator was upgraded to the *Advanced Driving Simulator*. The main difference to the previous design was the extension of the motion system in lateral direction, where a hydraulic cylinder realized a 5.6 m excursion. This modification improved considerably the quality of the simulator.

BMW BMW developed a 4 m high, hydraulic hexapod, with a small screen mounted onto the motion platform, together with a full-size car. This system was completely

rebuilt in 2003 and provided with a dome: now the driver enters the simulator through a tunnel/catwalk, to give the driver the idea he enters a car and not a simulator.

JARI, Nissan and WIVW Other 6-DOFs, hydraulic simulators were built in this period by JARI (1996) and Nissan (1999). The hexapod of IZVW/WIVW, built in 1999 is statically compensated using three additional cylinders. It is of interest that the hexapod doesn't seize the payload at the bottom, but at the height of the driver's head. The idea behind this concept is, that it is easier to tilt the payload around the driver's vestibular system, i.e. inside his head.

NADS-1 In 2002 the North American Driving Simulator (NADS-1) was presented at the University of Iowa. At that time it was by far the most advanced driving simulator. The NADS-1 is a 9-DOFs simulator, consisting of an xy-table on which a hexapod travels. On top of the hexapod, a turntable is mounted, which provides yaw-acceleration. A dome, with full-size car inside, rotates on top of the turntable.

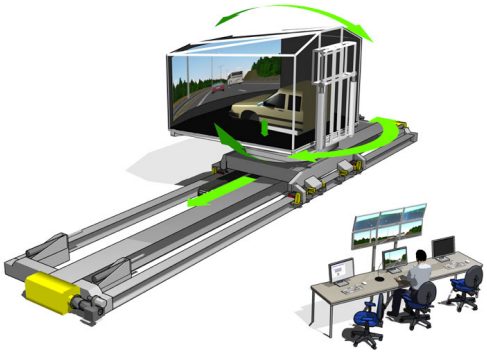
Renault In 2004 Renault introduced ULTIMATE (Dagdelen et al., 2009) with a hexapod on an xy-table, with a concept similar to NADS-1. The design of the motion system is carried out by BOSCH–REXROTH.

UoLDS The University of Leeds owns its own simulator since 2006 and claims to be one of the world's most advanced driving simulators in research environment. The research in which the simulator is currently employed, involves intelligent speed adaptation, effects of automated systems on safety and improved driver comprehension on traffic management signing.

Toyota In 2007 the NADS-1 simulator was exceeded in size by the Toyota Driving Simulator, built at Toyota's Higashifuji Technical Center in Susono City. The design is very similar to the NADS-1, but larger, and the main difference is found in the turntable. At the Toyota Driving Simulator, the car yaws inside the dome, whereas the NADS-1 yaws the entire dome, with the car inside of it. The simulator will be used for driving tests that are too dangerous to conduct in the real world, such as the effect of drowsiness, fatigue, inebriation, illness and inattentiveness.

In figure 2.3 pictures of some of the previously described simulator are reported.

There is still no consensus on which motion system design suits the demands of a realistic driving simulator best.



(a) VTI-III

(b) NADS-1

(c) ULTIMATE simulator

(d) UoLDS

(e) Toyota simulator.

Figure 2.3: Examples of existing simulators.

2.2 Classical Motion Cueing

Perception is the core of motion simulation. The human body has many inputs for motion perception: sight, hearing, vestibular and proprioceptive cues. While the first two categories are common to fixed simulators, the dynamic ones have to deal with the reproduction of the last ones too, in particular those acting on the *vestibular system*. Located in the inner ear (left and right), it is the prominent organ that provides information about linear and angular inertial accelerations of the body.

The Motion Cueing (MC) algorithm is the procedure that aims at best reproducing inside the simulator the stimuli that the driver would have perceived in the real car. The signals of the software-simulated vehicle are passed to the algorithm, and are elaborated to produce a reliable command for the actuators (in terms of perception quality). The algorithm is implemented keeping in mind that the platform is subject to physical limitations, hence the control signals have to be able to keep the device in safe conditions too. This last is called a *Washout Action*.

One of the first publications about Motion Cueing procedure is by Conrad and Schmidt (1971), where the basic setup for the classical approach was explained: since there, the main idea behind the algorithm is remained the same. A deeper description of a modern application of the classical MC is given by Reymond and Kemeny (2000); the logic block-scheme is in figure 2.4.

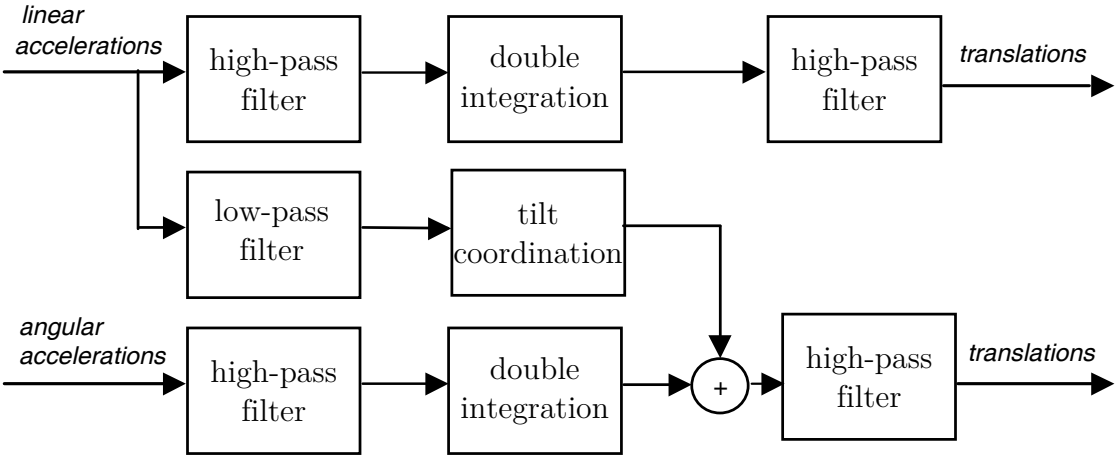


Figure 2.4: Classical Motion Cueing scheme.

The fundamental steps are

1. the software model of the vehicle dynamics produces the linear and rotational acceleration signals, that are fed to the algorithm;

2. both the linear and angular accelerations are *scaled*, to make the amplitude of the signals commensurate with the platform capabilities;

3. both the linear and angular accelerations are *high-pass* filtered (figure 2.5a). This operation eliminates the low frequency components of these signals. This is necessary because common simulators haven't the capabilities to reproduce the correspondent behaviours, for which a lot of space would be needed (with consequent difficulties in installation and exponential increase in terms of cost). Thus, the traditional approach is to keep only the high frequency behaviour, which is easier to reproduce in a limited space;

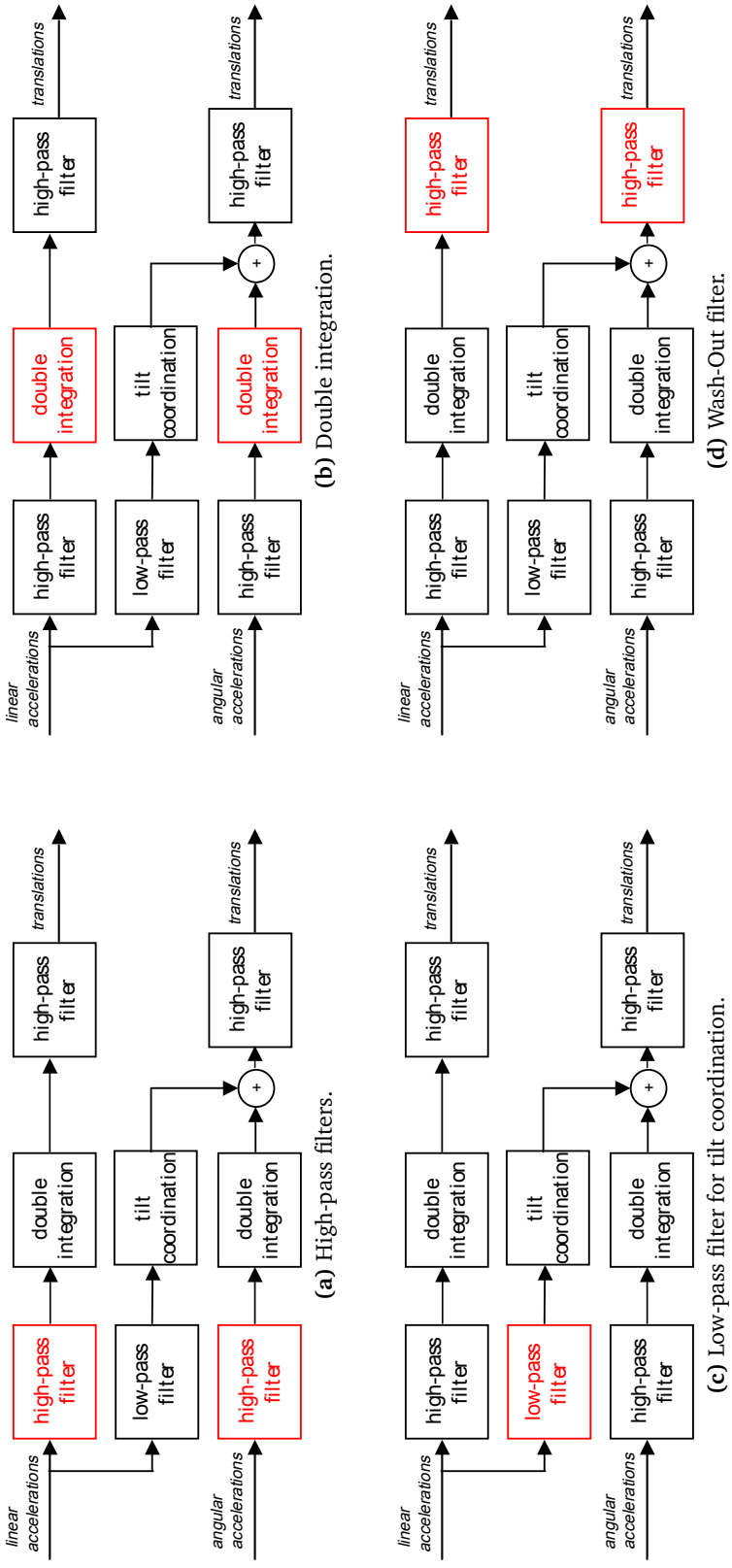
4. the resulting signals are *doubly integrated*, to obtain the correspondent spatial positions (figure 2.5b);

5. linear accelerations are *low-pass* filtered too. The correspondent part of the signal is then multiplied by a coefficient (usually the inverse of the gravity acceleration g) and summed to the angular position (figure 2.5c). This is the so-called *tilt-coordination*: the inertial effect of gravity is exploited to reproduce a part of the low frequency linear accelerations that the device wouldn't be able to mimic by itself;

6. the resulting position signals are again *high-pass* filtered (figure 2.5d). This time the task is to avoid undesired, divergent behaviours that may lead to infeasible position of the platform: it is the Washout Action, that keeps the platform as close as possible to the central, neutral position.

The procedure is based on a combination of passive filters, and the behaviour is regulated for each particular case (track, simulated car, driver's capabilities etc.) by tuning the filters parameters, namely gains and cut-off frequencies.

It is trivial to understand that visual and audio cues have to be synchronized with the inertial ones.



(a) High-pass filters.

(b) Double integration.

(c) Low-pass filter for tilt coordination.

(d) Wash-Out filter.

Figure 2.5: Different phases of classic MC algorithms.

2.3 Problems and limitations

The intrinsic structure of the classic approach shows many limitations and disadvantages that translate in possible low fidelity of perception signals and poor exploitation of the platform working space.

The high pass filters on accelerations (figure 2.5a) achieve the elimination of constant and/or slowly variable components of the signals, aiming to catch and reproduce only the fast variation, an affordable task in the limited space available. Despite being a reasonable solution, it is likely to introduce problems in the reliability of the signals. Consider the simulation of a turn: after the breaking event, the telemetry of longitudinal telemetry would show a fast decrease to negative values, and a subsequent slow (with respect to the breaking dynamics) increase to positive value when accelerating. The filtering effect could eliminate the slow component and set all the high frequency variations to take place around a mean value next to zero, hence the reproduced acceleration not only loses some components, but introduces mismatches that can heavily compromise the overall experience. An example of the situation is evident from figure 2.6. The inertial cues come into conflict with the visual ones and this situation can lead to *motion sickness*, in particular the so-called *simulator sickness* (Straus, 2005).

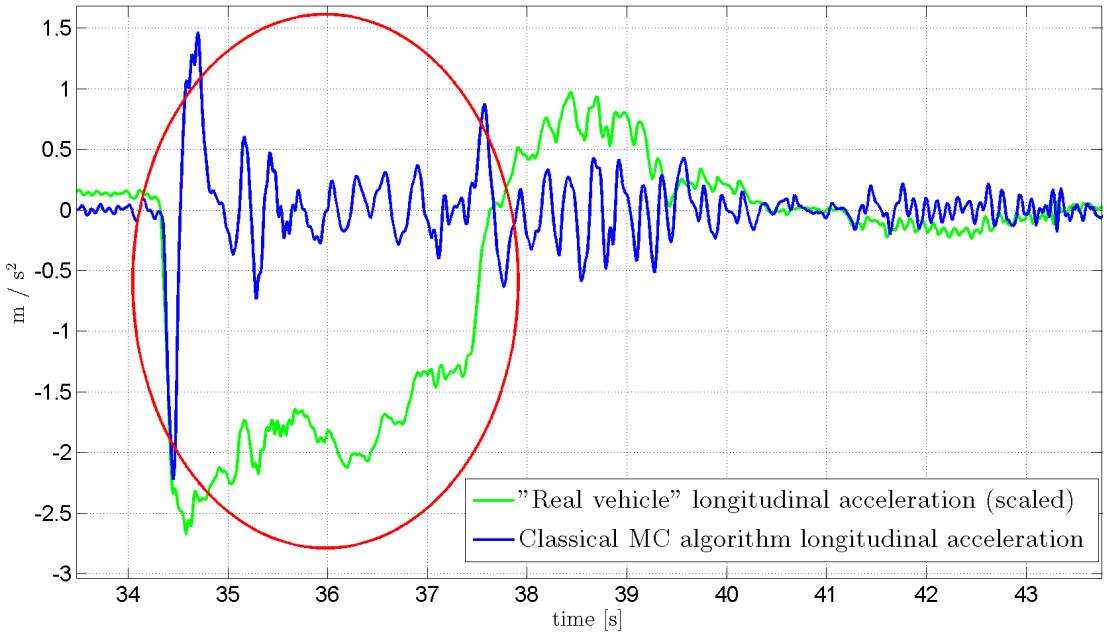


Figure 2.6: Classical Motion Cueing: *motion inversion*.

Simulator sickness, or cyber-sickness, defines possible maladies associated with simu-

lator usage and consist of feelings of nausea or discomfort. Simulator sickness may be triggered through *vection*, i.e. the perception of self-motion induced by visual stimuli, and occurs as a result of a disparity between visual and vestibular perceptual clues (Kennedy, Hettinger, and Lilienthal, 1988). Kennedy and Fowlkes (1992) characterize simulator sickness as "polysymptomatic", because several symptoms are at play, including blurred vision, cold sweating, concentration difficulty, confusion, drowsiness, eye strain, head fullness, nausea, pallor, and vomiting. Additional symptoms of cyber-sickness may include ataxia (postural disequilibrium or a lack of coordination), disorientation, dryness of mouth, fullness of stomach, headache, and vertigo. Since there appear to be several rather than one single source of these symptoms, Kennedy and Fowlkes (1992) define simulator sickness as "polygenic" too. Simulator sickness is sometimes classified as a form of motion sickness that may result from abrupt changes in movement or while the body's orientation is relatively fixed yet exposed to moving visual scenes. Note that this illness is very subjective and each human being is sensible to the problem in his own way: this problem has to be taken into account when dealing with the driver in the tuning phase.

Simulator sickness is therefore a problem to be avoided. On one hand, tilt coordination can help by introducing some low frequency features, on the other it is by itself not that easy to integrate: being obtained via low-pass filtering, it could introduce undesired delays. Moreover, this feature has to be handled so that the driver doesn't feel to be rotated, hence by keeping the rotations speed under a *perception threshold* (set to ± 3 degrees per second by Dagdelen, Reymond, Kemeny, Bordier, and Maïzi (2004)). By means of regulating only cut-off frequencies and gains, this is quite a hard task, and for this reason tilt coordination contribution is likely to be kept low.

Furthermore, after the double integration necessary to calculate the position of the platform, there is another high-pass filter for the Washout Action (figure 2.5d) whose aim is to assure the safety for the actuators. It may happen that the double integration determine a control signal that saturates the actuators, and this is unacceptable: beside the risks of damaging the device, a saturation will determine a non-natural behaviour thus compromising perceptive reliability. The rule of thumb in motion reproduction is that it is better not to have a certain cue, if there exists the risk that it will be a false cue, and this motivates the choice of a strategy that aims to keep the platform as close as possible to its neutral position, sacrificing working area exploitation.

All this elements make the tuning of the algorithm a challenging task: first of all, acting on filter parameters doesn't allow an easy physical correspondence between these regulations and the consequent behaviours. Then, it is intrinsically a conservative approach: it is impossible to include information about the limitations of the device, hence

the only reasonable choice is to impose a conservative behaviour.

During the years some modifications and improvements have been proposed. Parrish et al. (1975) proposed an *adaptive* strategy where the frequency parameters of the filters were modified after the minimization of a cost function of the kind

$$V_t = (r_t - a_t)^2 + w_1 \cdot v_t^2 + w_2 \cdot p_t^2 \tag{2.1}$$

which considers both the tracking error and the current positions and velocities; Sivan et al. (1982) propose an *optimal control* approach, based on the minimization of a function similar to (2.1). The last one is the first work which tries to integrate a *perception model*: the need for model-based solution to overcome all the stated problems is evident.

3

MPC Algorithm for Motion Cueing

In this chapter the derivation of the proposed algorithm will be illustrated. As specified, the chosen framework is *Model Predictive Control*. This methodology distinguished itself by other approaches in classical control theory for four cardinal elements

1. it is *model-based*, which is a great improvement in the information provided to the control algorithm;
2. it is an *optimal control* technique, where a cost function is minimized: in this sense, the control input is calculated in the best way possible, given the details of the problem, and can be easily adjusted to further variations;
3. it is a *constrained* optimal problem, so the constraints arising in the problem can be explicitly specified and handled;
4. it is a *predictive* strategy, so if a reference is know, with the help of the model the calculated input can be improved in term of control performance.

It will be shown that this control paradigm can be usefully exploited in the MC context, and even in the case when a reference signal is not available for prediction the other features will be crucial for the improvement of the performance of the algorithm, both for the final user (the driver) and the operator who will have to tune the algorithm to match the driver's requests

After an introduction of the generics about MPC and the chosen formulation, detailed information about the vestibular model and its derivation will be given, because of the great importance of this element in the procedure. Being a constrained optimal problem to be solved in real-time, the optimizer plays a central role and will be deeply illustrated. The tuning phase and the motivation behind the choices made will be analysed, showing how the constraints on the variables of the problem and the weights in the cost function have a central role.

3.1 Model Predictive Control

Model Predictive Control (MPC) is an advanced control technique widely used in industrial applications (Wang (2009); Maciejowski (2002)) since the 1980s. In recent years, robust and efficient implementations have been developed, as well as software tools in standard computational environments that ease the design of MPC algorithms. The main advantages of MPC can be summarized as follows:

- its underlying idea is simple and intuitive to understand;
- it's the only generic control technique that efficiently deals with constraints;
- it can handle Multi-Input Multi-Output (MIMO) systems without formally increasing the complexity of the problem;
- it can handle non linearities in both the model and the constraints.

3.1.1 MPC basics

In the MPC framework, it is common to consider a discrete-time problem; assume that at time $k \in \mathbb{Z}$ a reference trajectory $r(t|k)$, $t \geq k$ and a current measure of the output $y(k)$, are available. Note that the current input is *not yet computed*. Now, suppose to have a model of the process to be controlled and that the state of the system (or an estimate) is available. It is therefore possible to predict the future output

$$y(k+i|k), \quad i = 1, \ldots, N_p$$

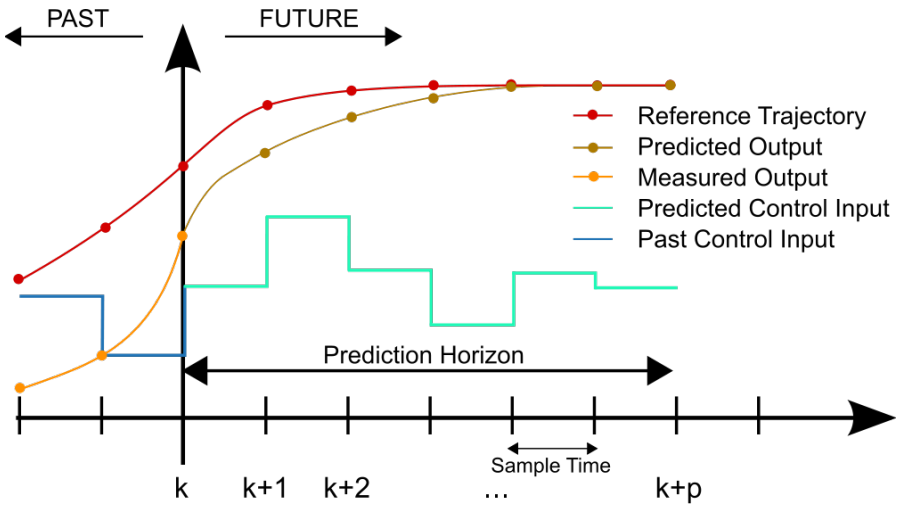


Figure 3.1: Representation of MPC principle.

corresponding to the input sequence

$$u(k+i|k), \quad i = 0, \ldots, N_p - 1$$

in a time-window of length N_p, where N_p is the prediction horizon length: figure 3.1 gives a graphical sketch of this (simple) principle.

The idea is to compute the input sequence $\hat{u}(k+i|k)$ that minimizes a *cost function*, which is typically related to the *tracking error*

$$\varepsilon(k+i|k) = r(k+i|k) - y(k+i|k), \quad i = 1, \ldots, N_p,$$

while respecting a set of *constraints* that can be on inputs, outputs, states (measured or estimated) and other quantities related to the problem. Once the optimal sequence is computed, the input to be applied at the current instant k is chosen as

$$u(k) = \hat{u}(k|k).$$

At time $k+1$, a new output $y(k+1)$ is measured and the algorithm is iterated applying again only the first element of the computed optimal input sequence. This procedure is known as *receding horizon*: at each time step, the prediction horizon is shifted in order to look N_p instants in the future, starting from the current time. In this way, iterating on the same prediction window and considering only the firs computed control, the algorithm can exploit the information coming from new measures and adapt its performance to correct errors due to exogenous disturbances, errors in the model, system variations and

so on, providing a certain degree of robustness: using the entire computed sequence before computing the new one, despite being much more effective from a computational point of view will clearly lead to a worse behaviour of the controller, and possibly to instability. As last remark, it is intuitive to understand that the receding horizon principle allows a "relaxation" of the predictive aspects of the MPC framework: even in absence of a reliable reference trajectory, the performance can be satisfactory if the control frequency is fast enough. This is a common case in industrial applications, where in absence of information about future references, constant signals are used (Maciejowski (2002)).

3.1.2 Process model

In the literature, different implementations of the MPC principle have been proposed, with different model structures. It is clear that the model is one of the crucial feature of the algorithm, and even if the receding horizon doesn't require a perfect model, the effectiveness of the control directly depends on how accurate the system representation is (particularly when reliable prediction is available).

Different kind of models have been proposed during the years

- FIR models
- step-response models
- impulse-response models
- transfer function models
- state-space models.

The first three classes are quite limited, complex and usable only for stable systems. Transfer functions are a better tool, but hard to manage when dealing with MIMO systems. *State-space models* adapt perfectly to the MPC paradigm: in linear form they have been widely study in control system theory, they make immediately accessible state variables and, if not directly available, are particularly well suited to design state estimators by using well-established tools of statistical filtering theory; more, they are the best way to represent MIMO systems. These reasons made this representation the most common choice in literature and real applications, including the Motion Cueing application we are considering in this dissertation.

Let's consider a linear, discrete state space model (sampled version of the continuous

process) of the form

$$\begin{aligned}\mathbf{x}_m(t+1) &= A_m\mathbf{x}_m(t) + B_m\mathbf{u}(t)\\ \mathbf{y}(t) &= C_m\mathbf{x}_m(t), \qquad\qquad t \in \mathbb{Z}.\end{aligned}$$

We assume that the input does not have an instantaneous effect on the output (*strictly proper* system). In our case, the MPC approach proposed by Wang (2009) has been adopted, hence we consider as the element to be optimized the *input difference*

$$\Delta\mathbf{u}(t) \triangleq \mathbf{u}(t) - \mathbf{u}(t-1).$$

This gives the advantage to explicitate and approximation of the *derivative* of the input signals, useful to include in the cost function as will be illustrated in the next section. Considering also the state difference

$$\Delta\mathbf{x}_m(t) \triangleq \mathbf{x}_m(t) - \mathbf{x}_m(t-1),$$

a new state equation can be written

$$\Delta\mathbf{x}_m(t+1) = A_m\Delta\mathbf{x}_m(t) + B_m\Delta\mathbf{u}(t).$$

Considering the output difference

$$\mathbf{y}(t+1) - \mathbf{y}(t) = C_mA_m\Delta\mathbf{x}_m(t) + C_mB_m\Delta\mathbf{u}(t)$$

and defining as augmented state

$$\mathbf{x}(t) \triangleq \begin{bmatrix}\Delta\mathbf{x}_m(t)^T & \mathbf{y}(t)^T\end{bmatrix}^T, \tag{3.1}$$

we obtain a new model

$$\mathbf{x}(t+1) = \left[\begin{array}{c|c} A_m & 0 \\ \hline C_mA_m & I \end{array}\right]\mathbf{x}(t) + \left[\begin{array}{c} B_m \\ \hline C_mB_m \end{array}\right]\Delta\mathbf{u}(t) \tag{3.2a}$$

$$\mathbf{y}(t) = \left[\begin{array}{c|c} 0 & I \end{array}\right]\mathbf{x}(t) \tag{3.2b}$$

where the control input is $\Delta\mathbf{u}(t)$.

3.1.3 Cost function

The cost function $J(t)$ chosen for the MC problem is *quadratic*, the most typical form in this context to avoid excessive complexity in resolution, and takes into account the error between the predicted trajectory and the future reference in the prediction window of size N_p, the future inputs and input difference in the *control horizon* N_c:

$$J(t) = \sum_{j=1}^{N_p} \delta(j)\|\mathbf{y}(t+j\,|\,t) - \mathbf{r}(t+j)\|^2 + \\ + \sum_{j=0}^{N_c-1} \lambda(j)\|\mathbf{u}(t+j)\|^2 + \sum_{j=0}^{N_c-1} \gamma(j)\|\Delta\mathbf{u}(t+j)\|^2, \quad (3.3)$$

$J(t)$ has to be minimized over $\mathbf{u}(t)$ and $\Delta\mathbf{u}(t)$. Observe that a term weighing $\Delta\mathbf{u}(t)$ is included in the cost function (3.3). In fact, as will be shown when describing the vestibular model, among the system inputs there are longitudinal accelerations, that are high-frequency, discontinuous signals. It is therefore convenient to act on their (approximate) derivative, to achieve a certain degree of regularity in the control signal, thus avoiding and excessive stress of the actuators, and possible unfeasibilities in the optimal problem.

Remark 3.1.1. Note that the control horizon N_c is distinguished from the prediction horizon N_p. The control horizon corresponds to the length of the control sequence that has to be calculated at each step, hence $N_c \leq N_p$ holds. If $N_c < N_p$, the last $N_p - N_c$ elements of the control sequence are considered constant and equal to the last computed element in the control horizon, i.e.

$$\mathbf{u}(t+N_c-1) = \mathbf{u}(t+N_c) = \mathbf{u}(t+N_c+1) = \cdots = \mathbf{u}(t+N_p) \quad (3.4)$$

which, expressed in terms of $\Delta\mathbf{u}$ as in (3.2), becomes

$$\Delta\mathbf{u}(t+N_c) = \Delta\mathbf{u}(t+N_c+1) = \cdots = \Delta\mathbf{u}(t+N_p) = 0. \quad (3.5)$$

In this way, the size of the optimal sequence to be computed is smaller thus easing the problem resolution. This is a common approach for the MPC framework, especially when the control frequency requires fast computing. The values of N_c and N_p have to be tuned carefully to get satisfactory performance; in the present work this approach has been adopted, with $N_c < N_p$ and it will be shown how the regulation of these parameters is an important aspect of the tuning phase of the whole algorithm.

3.1.4 Constraints

Constraints are very common in applied control. They could concern limits on actuators, on available space, on permitted dynamics, and so on. Model Predictive Control considers the presence of constraints: in this sense, it is a *constrained optimal* control procedure. Taking into account constraints in the formulation of a control strategy is an evident advantage, since it grants that if a solution for the optimisation problem exists, it will me a reliable control sequence. This is done at the cost of an increased complexity in finding the solution, since the introduction of constraints makes the problem impossible to be solved in an analytical way, and heuristic solvers have to be used. Beside the harder computational effort, the introduction of constraints leads to possible *infeasibility* of the problem: there could be a particular state of the system that makes impossible to find a solution, given a certain set of constraints. This aspect has to be taken in particular care when dealing with MPC, and the solver should be able to handle this critical case, for example by relaxing some of the constraints (the so called *soft* constraints, in opposition to *hard* ones which cannot be violated: see the discussion about *slack variables* in the book by Wang (2009)).

Constraints are usually set on inputs and outputs,

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}, \tag{3.6}$$

$$\mathbf{y}_{min} \leq \mathbf{y}(t) \leq \mathbf{y}_{max}; \tag{3.7}$$

given the formulation (3.2), it is appropriate to set constraints on input variations too

$$\Delta\mathbf{u}_{min} \leq \Delta\mathbf{u}(t) \leq \Delta\mathbf{u}_{max}. \tag{3.8}$$

It is possible also to set constraints on state space variables, but this case is not that common as the previous ones. This is due to the fact that in most problems, the value of the state isn't directly available and one has to rely on estimates: if the reliability of these estimates is somehow assured, then they can be used to evaluate constraints, in other cases it is better to avoid to introduce possible infeasibilities.

3.1.5 MPC as a Quadratic Programming problem

In the field of mathematical optimisation problems, *Quadratic Programming* (QP) problems have been widely studied. Different solving algorithm have been proposed, with advantages and disadvantages which have different impacts according to the characteristics of the problem involved. An optimisation problem can be defined a QP problem

if

1. it has a quadratic function to be minimized;
2. the constraints are linear.

The classic formulation of a QP problem is

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{x}^T c \tag{3.9}$$

$$A\mathbf{x} \leq b \tag{3.10}$$

with $\mathbf{x}, c, b \in \mathbb{R}^n$ and $H, A \in \mathbb{R}^{n\times n}$. H is a symmetric matrix; if it is *positive semidefinite*, function f is convex. In this case, if there exists a feasible solution $\mathbf{x}^*$ and f has a lower bound in the feasible region, there exists a global solution. If H is *positive definite* and there exists a feasible solution $\mathbf{x}^*$, then it is the unique global minimizer. For a more details on QP problems, see the book by Boyd and Vandenberghe (2004).

The advantage of dealing with QP problem is clear: it is well known in literature, not too difficult to solve and many optimizer are available (Wang and Boyd, 2010). Since the function (3.3) is quadratic, and the constraints (3.6),(3.7) and (3.8) are linear, the idea is to manipulate the MPC problem to convert it to a QP one. As proved in Wang (2009), it is always possible (and convenient) to reformulate (3.3) in order to obtain a generic Quadratic Programming problem where only $\Delta\mathbf{u}$ has to be minimized.

Let's consider system (3.2), the state evolution is

$$\begin{aligned}
&\mathbf{x}(t+1\,|\,t) = A\mathbf{x}(t) + B\Delta\mathbf{u}(t)\\
&\mathbf{x}(t+2\,|\,t) = A\mathbf{x}(t+1\,|\,t) + B\Delta\mathbf{u}(t+1) = A^2\mathbf{x}(t) + AB\Delta\mathbf{u}(t) + B\Delta\mathbf{u}(t+1)\\
&\qquad\vdots\\
&\mathbf{x}(t+N_p\,|\,t) = A^{N_p}\mathbf{x}(t) + A^{N_p-1}B\Delta\mathbf{u}(t) + A^{N_p-2}B\Delta\mathbf{u}(t+1) + \cdots +\\
&\qquad\qquad + A^{N_p-N_c}B\Delta\mathbf{u}(t+N_c-1),
\end{aligned} \tag{3.11}$$

considering $N_c < N_p$ hence (3.5). The corresponding outputs are

$$\begin{aligned}
\mathbf{y}(t+1\,|\,t) &= CA\mathbf{x}(t) + CB\Delta\mathbf{u}(t) \\
\mathbf{y}(t+2\,|\,t) &= CA\mathbf{x}(t+1\,|\,t) + CB\Delta\mathbf{u}(t+1) = CA^2\mathbf{x}(t) + CAB\Delta\mathbf{u}(t) + \\
&\qquad + CB\Delta\mathbf{u}(t+1) \\
&\vdots \\
\mathbf{y}(t+N_p\,|\,t) &= CA^{N_p}\mathbf{x}(t) + CA^{N_p-1}B\Delta\mathbf{u}(t) + CA^{N_p-2}B\Delta\mathbf{u}(t+1) + \cdots + \\
&\qquad + CA^{N_p-N_c}B\Delta\mathbf{u}(t+N_c-1). \qquad (3.12)
\end{aligned}$$

The outputs depends only on the current state $\mathbf{x}(t)$ and the input sequence $\Delta\mathbf{u}(t+i)$, $i = 0,\ldots,N_c-1$. If we rewrite (3.11) and (3.12) in vectorized form,

$$\mathbf{Y} = \mathrm{vec}\{\mathbf{y}(t+i\,|\,t)\}_{i=1,\ldots,N_p} = \begin{bmatrix} \mathbf{y}(t+1\,|\,t) \\ \mathbf{y}(t+2\,|\,t) \\ \vdots \\ \mathbf{y}(t+N_p\,|\,t) \end{bmatrix} \in \mathbb{R}^{(N_p\cdot n_{out})\times 1} \qquad (3.13)$$

$$\Delta\mathbf{U} = \mathrm{vec}\{\Delta\mathbf{u}(t+i\,|\,t)\}_{i=0,\ldots,N_c-1} = \begin{bmatrix} \Delta\mathbf{u}(t) \\ \Delta\mathbf{u}(t+1) \\ \vdots \\ \Delta\mathbf{u}(t+N_c-1) \end{bmatrix} \in \mathbb{R}^{(N_c\cdot n_{in})\times 1} \qquad (3.14)$$

with n_{out} and n_{in} the sizes of $\mathbf{y}$ and $\Delta\mathbf{u}$ respectively, the input-output evolution of the predictions can be represented as a single equation

$$\mathbf{Y} = F\mathbf{x}(t) + \Phi\Delta\mathbf{U} \qquad (3.15)$$

with matrices

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \quad \Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & & & & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B. \end{bmatrix} \qquad (3.16)$$

Considering a vectorization of the reference signal $\mathbf{r}(t+i)$, $i=1,\dots,N_p$ the cost function (3.3) becomes

$$J(\Delta\mathbf{U}) = (\mathbf{R}_s - \mathbf{Y})^T Q(\mathbf{R}_s - \mathbf{Y}) + \mathbf{U}^T S\mathbf{U} + \Delta\mathbf{U}^T R\Delta\mathbf{U} \tag{3.17}$$

where the matrices Q, S, R are block diagonals with elements corresponding to the weights in (3.3), hence they are *weight matrices*. Note that in most cases, these matrices are diagonal (the weights are typically constant, even if it possible without effort to specify variable values). The input sequence $\mathbf{u}(t+i)$ can be expressed as a function of $\Delta\mathbf{u}$ as

$$\underbrace{\begin{bmatrix}\mathbf{u}(t)\\ \mathbf{u}(t+1)\\ \vdots\\ \mathbf{u}(t+N_c-1)\end{bmatrix}}_{\mathbf{U}} = \underbrace{\begin{bmatrix} I & 0 & 0 & \cdots & 0\\ I & I & 0 & \cdots & 0\\ I & I & I & \cdots & 0\\ \vdots & & & & \vdots\\ I & I & I & \cdots & I\end{bmatrix}}_{T} \underbrace{\begin{bmatrix}\Delta\mathbf{u}(t)\\ \Delta\mathbf{u}(t+1)\\ \vdots\\ \Delta\mathbf{u}(t+N_c-1)\end{bmatrix}}_{\Delta\mathbf{U}} + \underbrace{\begin{bmatrix}\mathbf{u}(t-1)\\ \mathbf{u}(t-1)\\ \vdots\\ \mathbf{u}(t-1)\end{bmatrix}}_{\bar{\mathbf{U}}_i}. \tag{3.18}$$

By substituting (3.18) and (3.15) in (3.17), we obtain

$$\begin{aligned} J(\Delta\mathbf{U}) = (\mathbf{R}_s - F\mathbf{x}(t) - \Phi\Delta\mathbf{U})^T Q(\mathbf{R}_s - F\mathbf{x}(t) - \Phi\Delta\mathbf{U}) + \\ + \Delta\mathbf{U}^T R\Delta\mathbf{U} + (T\Delta\mathbf{U} + \bar{\mathbf{U}}_i)^T S(T\Delta\mathbf{U} + \bar{\mathbf{U}}_i); \end{aligned} \tag{3.19}$$

after some manipulations, and discarding the constant terms not dependent on $\Delta\mathbf{U}$ (J is a cost function, and these terms doesn't affect the minimization result), we get

$$J(\Delta\mathbf{U}) = \Delta\mathbf{U}^T(\Phi^T Q\Phi + R + T^T ST)\Delta\mathbf{U} + 2\Delta\mathbf{U}^T(\Phi^T Q(F\mathbf{x}(t) - \mathbf{R}_s) + T^T S\bar{\mathbf{U}}_i) \tag{3.20}$$

and calling

$$H \triangleq 2(\Phi^T Q\Phi + R + T^T ST), \tag{3.21}$$

$$c \triangleq 2(\Phi^T Q(F\mathbf{x}(t) - \mathbf{R}_s) + T^T S\bar{\mathbf{U}}_i) \tag{3.22}$$

equation (3.20) becomes

$$J(\Delta\mathbf{U}) = \frac{1}{2}\Delta\mathbf{U}^T H\Delta\mathbf{U} + \Delta\mathbf{U}^T c, \tag{3.23}$$

the classical expression of the cost function of a QP problem, as seen in (3.9), where the variable to minimize is $\Delta\mathbf{U}$.

The constraints should now be expressed as a function of $\Delta\mathbf{U}$ to obtain an analogous formulation to (3.10). Limitations on $\Delta\mathbf{u}(t)$ as in (3.8) are the easier to set, and become (Wang (2009))

$$\underbrace{\begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & I \\ -I & 0 & \cdots & 0 \\ 0 & -I & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & -I \end{bmatrix}}_{M_1} \underbrace{\begin{bmatrix} \Delta\mathbf{u}(t) \\ \Delta\mathbf{u}(t+1) \\ \vdots \\ \Delta\mathbf{u}(t+N_c) \end{bmatrix}}_{\Delta\mathbf{U}} \leq \underbrace{\begin{bmatrix} \Delta\mathbf{u}_{max} \\ \Delta\mathbf{u}_{max} \\ \vdots \\ \Delta\mathbf{u}_{max} \\ -\Delta\mathbf{u}_{min} \\ -\Delta\mathbf{u}_{min} \\ \vdots \\ -\Delta\mathbf{u}_{min} \end{bmatrix}}_{n_1}. \tag{3.24}$$

The constraints on $\mathbf{u}(t)$ shown in (3.6) are recontucted to $\Delta\mathbf{u}(t)$ by using equation (3.18),

$$\underbrace{\begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ I & I & \cdots & I \\ -I & 0 & \cdots & 0 \\ -I & -I & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ -I & -I & \cdots & -I \end{bmatrix}}_{M_2} \underbrace{\begin{bmatrix} \Delta\mathbf{u}(t) \\ \Delta\mathbf{u}(t+1) \\ \vdots \\ \Delta\mathbf{u}(t+N_c) \end{bmatrix}}_{\Delta\mathbf{U}} \leq \underbrace{\begin{bmatrix} \mathbf{u}_{max} - \mathbf{u}(t-1) \\ \mathbf{u}_{max} - \mathbf{u}(t-1) \\ \vdots \\ \mathbf{u}_{max} - \mathbf{u}(t-1) \\ -\mathbf{u}_{min} + \mathbf{u}(t-1) \\ -\mathbf{u}_{min} + \mathbf{u}(t-1) \\ \vdots \\ -\mathbf{u}_{min} + \mathbf{u}(t-1) \end{bmatrix}}_{n_2} \tag{3.25}$$

considering again the fact that at time t, all the elements computed before are known and assimilated to constant values. Nevertheless, the dependence on $u(t-1)$ makes matrix n_2 time-variant and has to be updated at each time step. The same holds true for the constraints on $\mathbf{y}(t)$: in fact, to transform the limitations (3.7) into constraints on $\Delta\mathbf{u}(t)$, equation (3.15) is exploited, giving

$$\underbrace{\begin{bmatrix} \Phi \\ -\Phi \end{bmatrix}}_{M_3} \Delta\mathbf{U} \leq \underbrace{\begin{bmatrix} \mathbf{Y}_{max} - F\mathbf{x}(t) \\ -\mathbf{Y}_{max} + F\mathbf{x}(t) \end{bmatrix}}_{n_3} \tag{3.26}$$

and here n_3 depends on the state $\mathbf{x}(t)$. $\mathbf{Y}_{min}$ ($\mathbf{Y}_{max}$) is the vectorization of the lower (upper) bounds $\mathbf{y}_{min}$ ($\mathbf{y}_{max}$) in (3.7), and is of size $N_p \cdot n_{out}$. By coupling equations (3.24), (3.25) and (3.26) we obtain the constraints in the classic form (3.10) for a QP problem,

$$\left[\begin{array}{c} M_1 \\ \hline M_2 \\ \hline M_3 \end{array}\right] \Delta\mathbf{U} \leq \left[\begin{array}{c} n_1 \\ \hline n_2 \\ \hline n_3 \end{array}\right]. \tag{3.27}$$

The described passages summarize as an MPC problem with quadratic cost function and linear constraints can be converted to a Quadratic Programming problem where the sequence $\Delta\mathbf{u}(t+i)$, $i = 0, \ldots, N_c - 1$ has to be calculated via minimization, for which a variety of solving algorithms are available in literature.

Remark 3.1.2. Constraints on state variables are less used, but sometimes can be useful: e.g., a mechanical system could have acceleration as output and position and velocity as state, and constraints on these latter quantities can be desirable. A simple and common way to set state constraints and to transform them in constraints on $\Delta\mathbf{U}$ as well, is to modify the model (3.62) in order to have as outputs the elements of the state that have to be constrained. In this way, (3.26) will automatically handle constraints on state variables, which have to be included in the vectors $\mathbf{Y}_{min}$ and $\mathbf{Y}_{max}$.

3.2 Vestibular Model

In the human being, the *(*perceptual system) is the set of biological systems and organs which are responsible for the capability of taking and elaborating information about the environment in order to recognize the physical situation and behave consequently. For what concerns the *motion perception*, three main systems systems cooperates

1. the *visual system*;
2. the *auditory system*;
3. the *vestibular system*.

In this work, the focus has been concentrated on the latter one: the integration with the first two will be investigated in further studies, and in particular for the visual cues, where works start to arise in the psychology field as in Pretto (2008)

The main function of the vestibular system (also know as *bony ear*) is to control motion and equilibrium by providing information about the transactional and rotational

accelerations that are acting on the body at the moment, including the instantaneous direction of the gravity force. It is located in the *inner ear* (figure 3.2 and is composed by two subsystems, the *semicircular canals* and the *otolithic organs*.

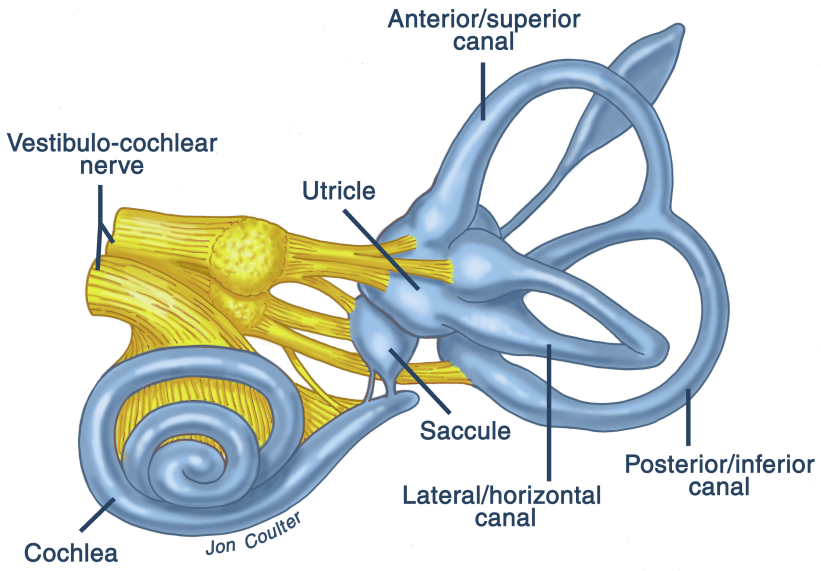


Figure 3.2: Structure of the inner ear.

3.2.1 The semicircular canals

The semicircular canals (from now on, abbreviated as SCC) are the organs of the vestibular system responsible for the perception of the rotations the body (in particular, the head) is subject to: they can be thought of as *angular accelerators* with a strong *damping* effect. Each ear has a set of three canals, as in figure 3.2:

1. the *horizontal* (or *lateral*) semicircular canal
2. the *anterior* (or *superior*) semicircular canal
3. the *posterior* semicircular canal.

Their spatial orientation is functional to their sensing purposes. For each ear, each of the three canals (meaning the plane it belongs to) is orthogonal to the other two; the horizontal canals form a 30 degree angle with the nasal-occipital plane, which may be considered as the “horizontal” plane when the eyes are pointing straight. The anterior and

posterior canals form angles of 45 degrees with the y-axis of the head, whose direction is from the back to the front of the head: figure 3.3 graphically explain the typical situation described. Note that, having to work together, there is a specific relative positioning of the canals among the two ears: in fact, while both the horizontal canals are coplanar (having to sense the yaw rotation), the left anterior canal is coplanar with the right posterior one and vice-versa (pitch and roll rotations). The canals themselves are elliptical, hollow

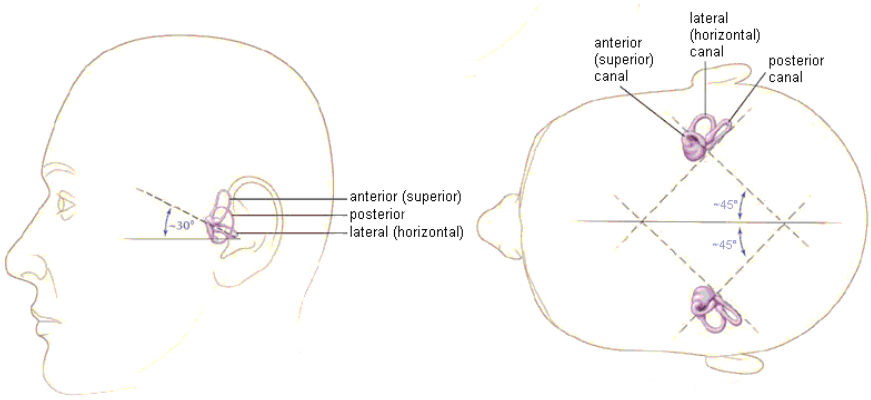


Figure 3.3: Spatial and relative positioning of the SCC.

arcs with about 240 degree of extension, filled with a fluid known as endolymph. The three canals of each ear have a common base, the *utricle* (which is part of the otholitic system); at one of the junctures of each canal with the utricle, the canal cavity swells to form a bulbous expansion called the *ampulla* which contains the sensory ephithelium or crista. The crista contains bundles of sensory hair cells that extend into a gelatinous mass called the cupula. The cupula bridges the width of the ampulla cavity, forming a seal through which endolymph cannot circulate. When the head turns in the plane of one of the canals, the inertia of the endolymph produces a force across the cupula, deflecting it in the opposite direction of head movement and causing a displacement of the hair bundles in each hair cell. This hair is linked through the crista to the vestibular nerve, which eventually transfers the information to the central nervous system. A graphical explanation is depicted in figure 3.4.

3.2.1.1 Mathematical model

It is clear that a mathematical modelization of such a complex human system is a challenging task; researches has seen a great incentive during the '60s and '70s in the context of the aero-spatial missions, and during the following years the aerospace field has been the leader of such studies. Hence, the mathematical model derived for the MC problem

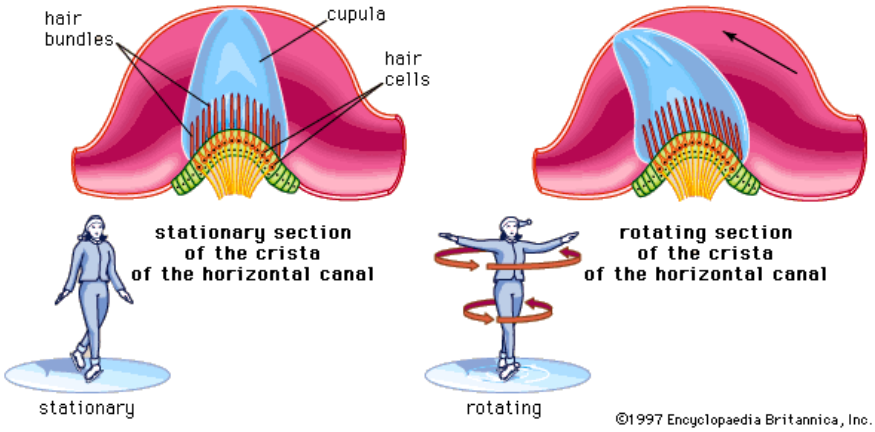


Figure 3.4: Mechanism of the motion perception in the SCC.

at hand derives from a thorough review of all the literature.

The classic starting point is the work by Zacharias (1978), which is a summary of all the major works until 1978. Zacharias reported that Steinhausen (1971) first developed a linear, second-order model of canal dynamics to explain the observed characteristics of vestibular-induced eye movements in fish (pike). This model was further refined by the "torsion-pendulum" model of Van Egmond, Groen, and Jongkees (1949). The transfer function for this overdamped system is the following

$$\frac{\delta(s)}{\omega(s)} = \frac{Ks}{(1+s\tau_s)(1+s\tau_l)}, \tag{3.28}$$

that relates the rotation of the head ω to the deflection δ of the cupula, which is then supposed to be proportional to the perceived rotation, namely $\hat{\omega}$ (note that the *hat* will indicate the perceived quantities from now on). Note that this model, where the dynamics is substantially a band-pass filter similar to a classical Washout filter, has been derived by evaluating the rotations around the z-axis only. Different values have been proposed during those years for the time constants, and the difference can be attributed, in general, to the different measurement method, which could be based on subjective evaluation or on the measurement of the nystagmus, i.e. the involuntary eye movement. This highlights once more the connection between the vestibular and visual perceptive systems. Further studies showed that the torsion-pendulum model does not completely represent rotational sensation, in particular step variations in the accelerations would be seen by this model as constant, while in practice they decay to zero (these human sensors are rather "differential" sensors than "absolute"). Young and Oman (1969) formulated an adaptation operator and cascaded it with the torsion-pendulum model to resolve

the conflicts between the response predicted by the torsion-pendulum model and the perceptual responses measured in experiments, obtaining

$$\frac{\delta(s)}{\omega(s)} = \frac{Ks}{(1+s\tau_s)(1+s\tau_l)} \frac{s\tau_a}{1+s\tau_a}. \tag{3.29}$$

Zacharias (1978) reported several experiments suggesting an additional *lead* component, that is substantially a first order high-pass term, $1+s\tau_L$, that represents the phase lead behaviour. With the addition of this component, Zacharias elaborated a new model which assumed that the perceived angular velocity $\hat{\omega}$ is proportional to the *afferent firing rate* (AFR) of the vestibular nerve, i.e. to the neural transduction dynamics, whose logic scheme is reported in figure 3.5.

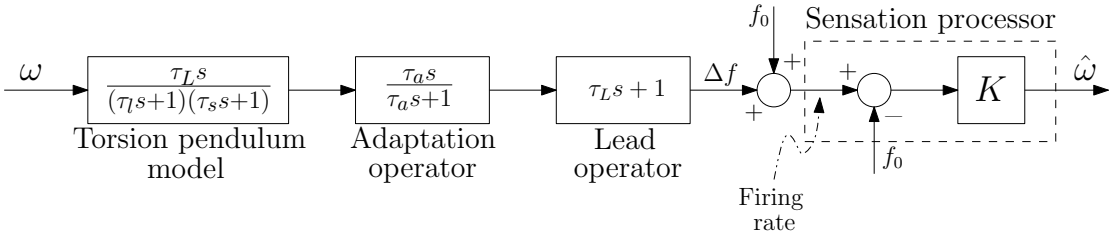


Figure 3.5: Block scheme considering the AFR dynamics of the vestibular nerve.

For what concerns the time constants, τ_s, based on the work by Steer (1967), has to be chosen based on the theoretic, hydrodynamical properties of the semicircular canals and in any case fast enough; τ_l (slow constant) is more clearly defined, based on the nystagmus measurement, while τ_L is harder to define a priori and should need to be adapted to the personal characteristics of the user.

In more recent years, the analysis has been reprised by Telban and Cardullo (Telban, Wu, Cardullo, and Center (2000); Telban, Cardullo, and Houck (2005)), with the aim of deriving more reliable values for the parameters, thanks to modern measurement methods. The starting point is substantially the same transfer function as (3.29) (considering the lead term), but relating the cupula deflection with the angular acceleration α

$$\frac{\delta_c(s)}{\alpha(s)} = K_{SCC} \frac{s\tau_a}{1+s\tau_a} \frac{1+s\tau_L}{(1+s\tau_1)(1+s\tau_2)}. \tag{3.30}$$

The values of the constants have been calculated by different scientists, and the work of Fernandez and Goldberg (1971) has great importance as a starting point for Telban and Cardullo: by directly measuring the response to variations in amplitude and frequency on the input acceleration signals of the afferent firing rate of the vestibular nerve for the

squirrel monkey, they found the values in (3.31)

$$\frac{AFR(s)}{\alpha(s)} = 3.44\frac{80s}{1+80s}\frac{1+0.049s}{(1+5.7s)(1+0.003s)}. \tag{3.31}$$

Fernandez and Golderbg find out that a more accurate value of τ_2 for man is 0.005. Telban and Cardullo started from (3.31) and derived the model in (3.32)

$$\frac{\hat{\omega}(s)}{\alpha(s)} = 5.73\frac{80s}{1+80s}\frac{1+0.06s}{(1+5.73s)(1+0.003s)}. \tag{3.32}$$

For the purposes of this work, equation (3.32) is modified to have as input the real rotational velocity, as in (3.33)

$$\frac{\hat{\omega}(s)}{\omega(s)} = 5.73\frac{80s^2}{1+80s}\frac{1+0.06s}{(1+5.73s)(1+0.003s)}. \tag{3.33}$$

To avoid numerical problem, the term relative to the fast constant τ_2 is eliminated, and to maintain the realizability of the transfer function the lead term $(1+0.06s)$ is discarded too, having a dynamics near to the chosen control frequency (100 Hz). The final transfer function used for each one of the canals is (3.34)

$$W_{SCC}(s) = \frac{\hat{\omega}(s)}{\omega(s)} = 5.73\frac{80s^2}{(1+80s)(1+5.73s)}; \tag{3.34}$$

figure 3.6 reports its frequency representation.

3.2.2 The otolith organs

The *maculae* (i.e. the otolithic systems) are the elements of the vestibular system that provide linear motion sensation in humans for specific force, meaning the combination of transactional acceleration and gravity force (whose interactions will be clarified when explaining the tilt-coordination effect). Each inner ear has two maculae, the utricle (common base for the canals) and the saccule (extension of the utricle). These organs are composed by a slightly curved bony base where the sensory cells lie (*sensory epithelium*), covered by a gelatinous membrane where the receptive cilia of the sensory cells are dunked. Over this *gelatinous layer* there is the *otoconial layer*, a membrane made of a viscous gel and covered by the otholits, which are basically calcium carbonate crystals. The whole structure stick out from the macular cavity, filled with endolymph. The sensing process is based on the different inertia between the otoliths and the gel: a linear movement makes the otoliths move against the cilia, connected to the vestibular nerve sending the sensory

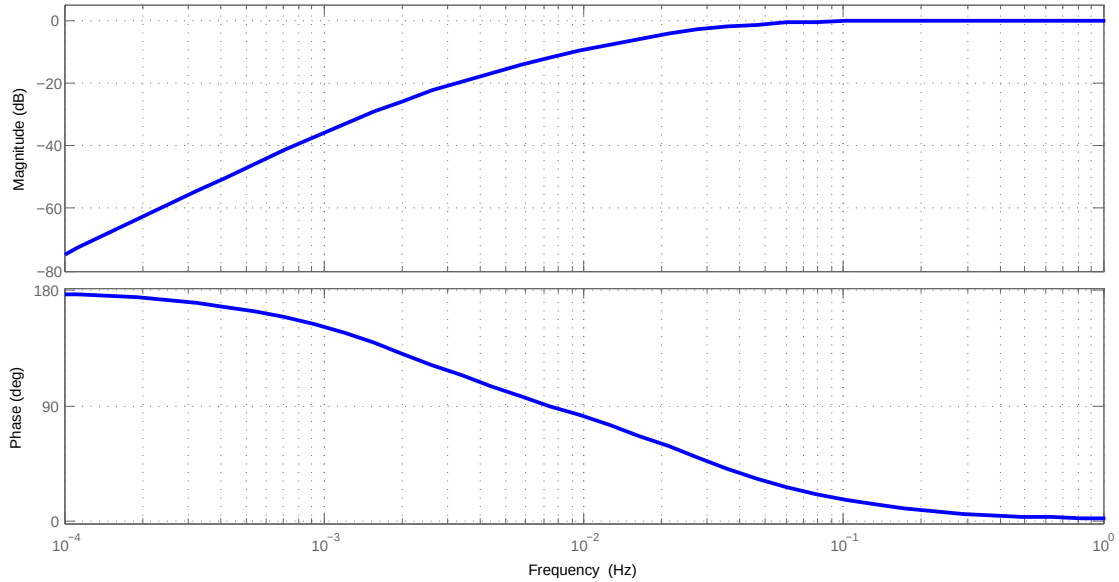


Figure 3.6: Bode diagram of SCC transfer function.

signal. Figure 3.7 depicts the structure of the organ. According to their spatial disposition, the utricle and the saccule sense the horizontal and vertical movement, respectively: this organs are coplanar between the two ears.

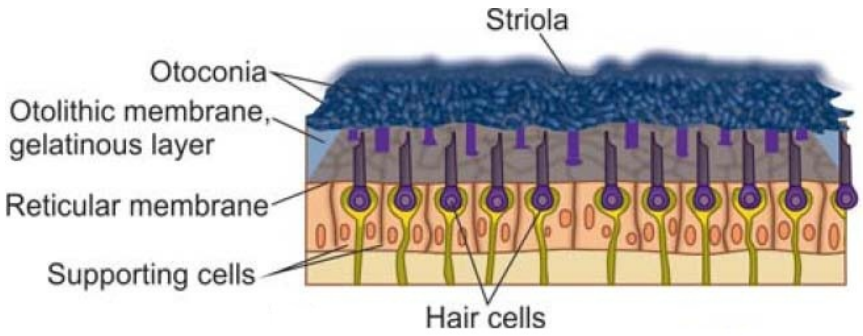


Figure 3.7: Structure of the otoliths.

3.2.2.1 Mathematical model

At the time of Zacharias (1978) work there was not a large body of literature available on the modelization of otoliths, since it was harder than in the semicircular canals case to understand which were the stimuli they were responsible to perceive. In particular, many test were carried on by focusing only on tilting the subject, hence considering only static inclinations with respect to the vertical axis (gravity). Moreover, it wasn't clear if

the perceived transients were on accelerations or velocities. One of the first work was by Meiry (1965), who first investigated *subjective* responses to linear motion by using a cart to produce longitudinal sinusoidal motion. By measuring the subjective indication of direction, he obtained a transfer function relating perceived velocity $\hat{v}$ to stimulus velocity v:

$$\frac{\hat{v}(s)}{v(s)} = \frac{Ks\tau_L}{(1+s\tau_L)(1+s\tau_S)} \tag{3.35}$$

The basic structure is the torsional pendulum as for semicircular canals. Young and Meiry (1968) noted that the model (3.35) failed to predict the otoliths' response to sustained tilt angle as indicated by behavioral and physiological data. They proposed a revised model of specific force sensation (hence supposing that the acceleration, and not the velocity is perceived) with a smaller long time constant and an additional lead term, thus modeling both perceived tilt and acceleration in response to acceleration input. This model presumes the equivalence of linear acceleration sensation with that of tilt. A block scheme representation is in figure 3.8.

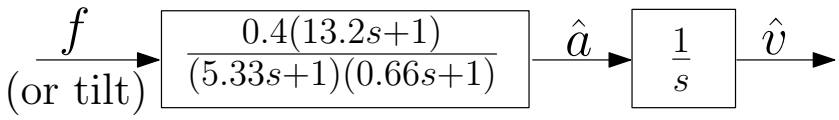


Figure 3.8: Block scheme for new Meiry and Young otolithic model.

After these first researches, Grant and Best obtained the transfer function (3.36) relating the relative displacement of the otoconial layer with respect to the head, x, with the force input f, through the density of the endolymph ρ_e and the density of the otoconial membrane ρ_0

$$\frac{x(s)}{f(s)} = \left(1 - \frac{\rho_e}{\rho_0}\right) \frac{\tau_1\tau_2}{(1+s\tau_1)(1+s\tau_2)}; \tag{3.36}$$

they estimated τ_2 during their experiments by measuring the maximum displacement of the otoliths in response to step variation in linear velocity. The otolith mechanical structure and the consequent dynamics of the afferent firing rate was proposed by Orsmby, who introduced the system (3.37), using a Wiener-Hopf equation to find the correct parameters

$$\frac{\hat{f}(s)}{f(s)} = \underbrace{\frac{Bs+(B+C)A}{s+A}}_{\text{Mechanics and AFR}} \cdot \underbrace{\frac{K(s+A)}{(s+F)(s+G)}}_{\text{Higher Centers Dynamics}}; \tag{3.37}$$

note that the higher centres correspond to the area of the cerebral cortex where the sensory

information are elaborated. Fernandez and Goldberg (1976) studied the discharge of peripheral otolith neurons in response to sinusoidal force variations in the squirrel monkey, thus estimating parameters for their transfer function; their work has been later refined by Hosman and Van der Vaart (1978), who proposed the transfer function

$$\frac{AFR(s)}{f(s)} = 33.3\frac{1+s}{(1+0.5s)(1+0.016s)}. \tag{3.38}$$

Telban et al. (2005) refined the model (3.38) and proposed the model

$$W_{OTO}(s) = \frac{\hat{f}(s)}{f(s)} = 0.4\frac{1+10s}{(1+5s)(1+0.016s)}, \tag{3.39}$$

the chosen one in the present work. In figure 3.9 the Bode diagram is reported.

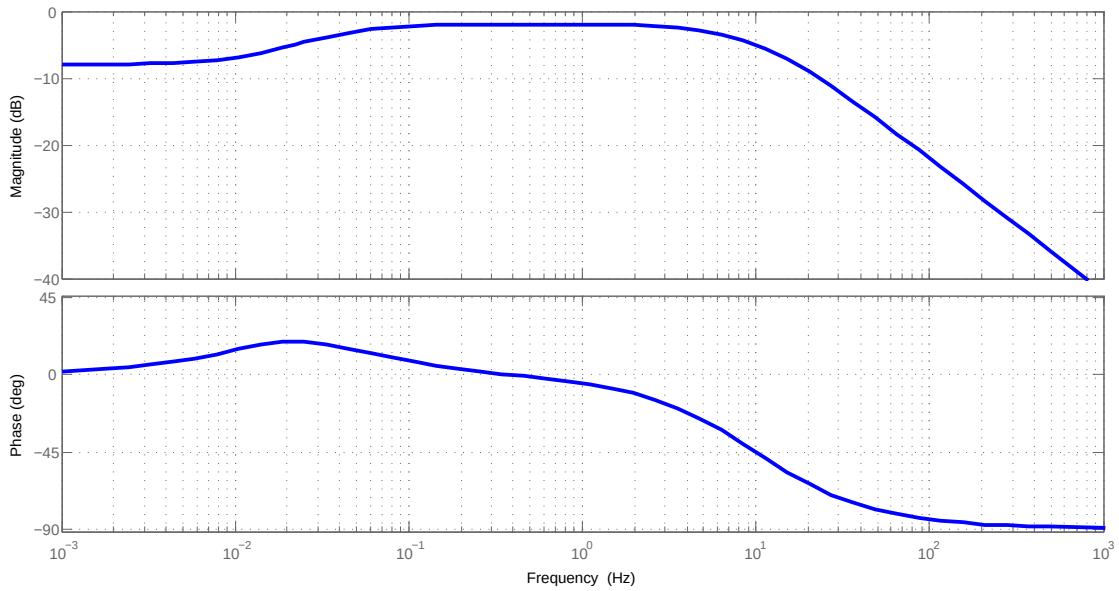


Figure 3.9: Bode diagram of otoliths transfer function.

3.2.3 Tilt coordination

An important component of perception in a dynamic simulator is given by *tilt coordination*. Otoliths are incapable to discriminate between gravitational and longitudinal forces without the help of other cues (like the ones coming from visual system). Hence, by using a non-zero pitch angle and without any other visual reference, it is possible to provide the driver in the simulator with a “fake” longitudinal acceleration sensation. The same holds true for roll and lateral acceleration. Such approach goes under the name of

tilt coordination. Taking into account this effect is crucial to reproduce low frequency behaviour with a reduced range working area. This effect is summarized in figure 3.10: it's easy to understand how the effect of the gravity component, in presence of a non-zero pitch angle θ, contributes to a greater perception of longitudinal acceleration (in absence of visual information). Tilt coordination is particularly relevant when predictions of the

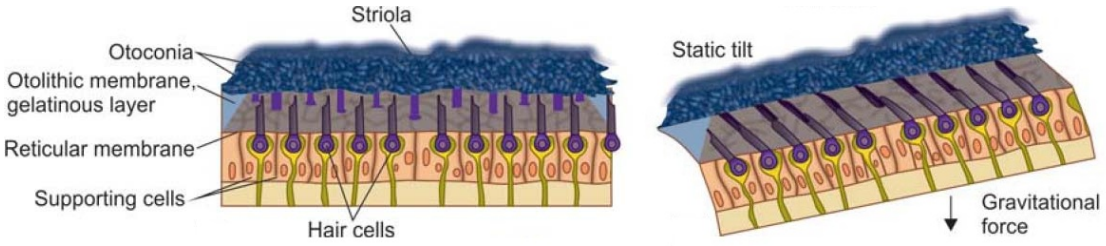


Figure 3.10: Effect of tilt coordination on the otoliths.

future vehicle trajectories are available, since, differently from classical cueing strategies, it can yield to less conservative strategies and thus to better exploitation of the platform working space. In the perception model, because of the linearisation, tilt coordination is nothing but a further contribution in the otoliths model $W_{OTH}(s)$ due to the pitch angle in the longitudinal direction and to the roll angle in the lateral direction.

With a mathematical formalization, given a pitch angle θ and a roll angle ϕ, one can apply a 2-DOFs rotation to the gravity vector $\mathbf{g} = [\,0\;0\;g\,]^T$ to obtain its components in the platform's coordinate system

$$\mathbf{g}_{PLAT} = R_x(\theta)R_y(\phi)\mathbf{g} = \begin{bmatrix} -g\sin\theta \\ g\cos\theta\sin\phi \\ g\cos\theta\cos\phi \end{bmatrix}, \tag{3.40}$$

where the subscript or the rotation matrix R indicates the axis around which the rotation is applied. Now, if the driver should be provided an acceleration $\mathbf{a} = [\,a_x\;a_y\;a_z\,]^T$, given (3.40) it suffices to generate the *specific acceleration* $\tilde{\mathbf{a}} = \mathbf{a} - \mathbf{g}_{PLAT}$

$$\tilde{\mathbf{a}} = \begin{bmatrix} \tilde{a}_x \\ \tilde{a}_y \\ \tilde{a}_z \end{bmatrix} = \begin{bmatrix} a_x + g\sin\theta \\ a_y - g\cos\theta\sin\phi \\ a_z - g\cos\theta\cos\phi \end{bmatrix} \approx \begin{bmatrix} a_x + g\theta \\ a_y - g\phi \\ a_z - g \end{bmatrix}, \tag{3.41}$$

using the small-angle linearisation, suitable for the application at hand.

3.2.4 State space realization of the complete vestibular model

In order to use the perception models in the MPC approach, $W_{OTO}(s)$ and $W_{SCC}(s)$ need to be converted in state space form and coupled with the tilt coordination contribution for all the 6 DOFs.

Recalling the previously derived model, the transfer functions for semicircular canals and otoliths are, respectively

$$W_{SCC,i}(s) = \frac{\hat{\omega}_i(s)}{\omega_i(s)} = 5.73\frac{80s^2}{(1+80s)(1+5.73s)} = \frac{s^2}{s^2 + s\left(\frac{1}{T_a}+\frac{1}{T_L}\right)+\frac{1}{T_aT_L}} \tag{3.42}$$

$$W_{OTO,i}(s) = \frac{\hat{f}_i(s)}{f_i(s)} = 0.4\frac{1+10s}{(1+5s)(1+0.016s)} = \frac{K\tau_a}{\tau_l\tau_s}\frac{s+\frac{1}{\tau_a}}{s^2 + s\left(\frac{1}{\tau_s}+\frac{1}{\tau_L}\right)+\frac{1}{\tau_s\tau_L}} \tag{3.43}$$

with the parameters highlighted. The transfer functions are the same for all the 6 DOFs, hence the subscript i stands for x, y, z. Following classic realisation theory results, the corresponding matrices are

$$A_{SCC,i} = \begin{bmatrix} -\left(\frac{1}{T_a}+\frac{1}{T_L}\right) & 1 \\ -\frac{1}{T_aT_L} & 0 \end{bmatrix}, \qquad B_{SCC,i} = \begin{bmatrix} -\left(\frac{1}{T_a}+\frac{1}{T_L}\right) \\ \frac{1}{T_aT_L} \end{bmatrix}, \tag{3.44a}$$

$$C_{SCC,i} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D_{SCC,i} = 1 \tag{3.44b}$$

for the semicircular canals with $i = x, y, z$ indicating rotations around x-axis (roll angle ϕ), y-axis (pitch angle θ) and z-axis (yaw angle ψ), respectively; and

$$A_{OTO,i} = \begin{bmatrix} -\left(\frac{1}{\tau_s}+\frac{1}{\tau_L}\right) & 1 \\ -\frac{1}{\tau_s\tau_L} & 0 \end{bmatrix}, \quad B_{OTO,i} = \begin{bmatrix} \frac{K\tau_a}{\tau_l\tau_s} \\ \frac{K}{\tau_l\tau_s} \end{bmatrix}, \quad C_{OTO,i} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D_{OTO,i} = 0 \tag{3.45}$$

for the otolithic system, with i indicating the x, y, z axes.

By coupling (3.44) along the rotational DOFs,the complete model for the semicircular canals is obtained

$$A_{SCC} = \text{blkdiag}\{A_{SCC,x}, A_{SCC,y}, A_{SCC,z}\} \tag{3.46a}$$

$$B_{SCC} = \text{blkdiag}\{B_{SCC,x}, B_{SCC,y}, B_{SCC,z}\} \tag{3.46b}$$

$$C_{SCC} = \text{blkdiag}\{C_{SCC,x}, C_{SCC,y}, C_{SCC,z}\} \tag{3.46c}$$

$$D_{SCC} = \text{blkdiag}\{D_{SCC,x}, D_{SCC,y}, D_{SCC,z}\}, \tag{3.46d}$$

where the input signals are the chassis angular velocities $\mathbf{u}_{SCC} = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T$ and the output signals $\mathbf{y}_{SCC} = [\hat{\dot{\phi}}\ \hat{\dot{\theta}}\ \hat{\dot{\psi}}]^T$ are the perceived angular velocities.

Similarly, by coupling (3.43) in the same way as (3.46) we obtain the complete otholitic system matrices

$$A_{OTO} = \text{blkdiag}\{A_{OTO,x}, A_{OTO,y}, A_{OTO,z}\} \tag{3.47a}$$
$$B_{OTO} = \text{blkdiag}\{B_{OTO,x}, B_{OTO,y}, B_{OTO,z}\} \tag{3.47b}$$
$$C_{OTO} = \text{blkdiag}\{C_{OTO,x}, C_{OTO,y}, C_{OTO,z}\} \tag{3.47c}$$
$$D_{OTO} = \text{blkdiag}\{D_{OTO,x}, D_{OTO,y}, D_{OTO,z}\}, \tag{3.47d}$$

with the chassis longitudinal accelerations as inputs ($\mathbf{u}_{OTO} = [a_x\ a_y\ a_z]^T$) and the perceived longitudinal forces as outputs ($\mathbf{y}_{OTO} = [\hat{a}_x\ \hat{a}_y\ \hat{a}_z]^T$). Note that in (3.43) inputs and outputs are forces, but the transfer functions (and the consequent state space model) are clearly identical for accelerations, which differ from forces by the (constant) value of the mass m. To consider the effect of tilt-coordination, (3.41) can be rewritten in matricial form as

$$\tilde{\mathbf{a}} = H \cdot \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\beta} \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & g & 0 \\ 0 & 1 & 0 & -g & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.48}$$

with $\boldsymbol{\beta} = [\phi\ \theta\ \psi]^T$. To use the angular rates as inputs to the otholit system (as requested by the final, complete vestibular model), the *augmented state*

$$\mathbf{x}_{\overline{OTO}} = \begin{bmatrix} \mathbf{x}_{OTO} \\ \boldsymbol{\beta} \end{bmatrix} \tag{3.49}$$

has to be introduced. The corresponding augmented system is
$\Sigma_{\overline{OTO}} = \{A_{\overline{OTO}}, B_{\overline{OTO}}, C_{\overline{OTO}}, D_{\overline{OTO}}\}$, with

$$A_{\overline{OTO}} = \left[\begin{array}{c|c} A_{OTO} & \bar{B} \\ \hline 0 & 0 \end{array}\right] \qquad B_{\overline{OTO}} = \left[\begin{array}{c|c} B_{OTO} & 0 \\ \hline 0 & I_3 \end{array}\right] \tag{3.50a}$$
$$C_{\overline{OTO}} = \left[\begin{array}{c|c} C_{OTO} & 0 \end{array}\right] \qquad D_{\overline{OTO}} = \left[\begin{array}{c|c} D_{OTO} & 0 \end{array}\right], \tag{3.50b}$$

with $\mathbf{u}_{\overline{OTO}} = [\mathbf{u}_{OTO}^T\ \mathbf{u}_{SCC}^T]^T = [\mathbf{a}^T\ \dot{\boldsymbol{\beta}}^T]^T$ as inputs, $\mathbf{y}_{\overline{OTO}} = \mathbf{y}_{OTO} = [\hat{a}_x\ \hat{a}_y\ \hat{a}_z]^T$ as outputs

and from (3.41)

$$\bar{B} = B_{OTO} \cdot \begin{bmatrix} 0 & g & 0 \\ -g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.51}$$

System (3.50) is an appropriate, linear representation of the vestibular system relating real accelerations and rotational velocities to the corresponding perceived ones. For our purposes it is convenient to make the *positions* and *velocities* of the eye-point[1] explicit. This gives two advantages

1. typical motion controllers of driving simulators act on the position of the eye-point, hence they have to be provided with those coordinates;

2. it is useful to have this physical values in the MPC algorithm to take advantage of imposing constraints on velocities and positions as well as suitable weights in the cost function.

This explicitation can be done by using a simple "integral system",

$$\dot{\mathbf{x}}_I = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_I + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{u}_I = A_I \mathbf{x}_I + B_I \mathbf{u}_I, \tag{3.52}$$

with state and input vectors

$$\mathbf{x}_I = \begin{bmatrix} p_x & v_x & p_y & v_y & p_z & v_z \end{bmatrix}^T, \qquad \mathbf{u}_I = \mathbf{u}_{OTO} \tag{3.53}$$

where p_i and v_i are positions and velocities along the i-th axis, respectively.

[1] The *eye-point* is intended to be the reference spatial point where the platform motion is applied; in this dissertation, it corresponds to the position of the head of the driver, where the vestibular system is located.

The complete vestibular system is then $\Sigma_V = \{A_V, B_V, C_V, D_V\}$, where

$$A_V = \begin{bmatrix} A_{SCC} & 0 & 0 \\ 0 & A_{\overline{OTO}} & 0 \\ 0 & 0 & A_I \end{bmatrix} \qquad B_V = \begin{bmatrix} 0 & B_{SCC} \\ B_{\overline{OTO}} & \\ B_I & \end{bmatrix} \tag{3.54a}$$

$$C_V = \begin{bmatrix} C_{SCC} & 0 & 0 \\ 0 & C_{\overline{OTO}} & 0 \\ 0 & 0 & I \\ & 0 & \end{bmatrix} \qquad D_V = \begin{bmatrix} 0 & D_{SCC} \\ D_{\overline{OTO}} & \\ 0 & \\ 0 & I_3 \end{bmatrix} \tag{3.54b}$$

and the state, input and output vectors are, respectively,

$$\mathbf{x}_V = \begin{bmatrix} \mathbf{x}_{SCC}^T & \mathbf{x}_{\overline{OTO}}^T & \mathbf{x}_I^T \end{bmatrix}^T \in \mathbb{R}^{21} \tag{3.55}$$

$$\mathbf{u}_V = \begin{bmatrix} a_x & a_y & a_z & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \in \mathbb{R}^6 \tag{3.56}$$

$$\mathbf{y}_V = \begin{bmatrix} \mathbf{y}_{SCC}^T & \mathbf{y}_{\overline{OTO}}^T & \beta^T & \mathbf{x}_I^T & \dot{\beta}^T \end{bmatrix}^T \in \mathbb{R}^{18}, \tag{3.57}$$

$$\mathbf{y}_{SCC} = \begin{bmatrix} \hat{\dot{\phi}} & \hat{\dot{\theta}} & \hat{\dot{\psi}} \end{bmatrix}^T \qquad \mathbf{y}_{SCC} = \begin{bmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \end{bmatrix}^T. \tag{3.58}$$

Finally, the system is discretized with a zero-order old method, thus obtaining

$$\begin{aligned} \mathbf{x}_{VEST}(t+1) &= A_{VEST}\mathbf{x}_{VEST}(t) + B_{VEST}\mathbf{u}(t) \\ \mathbf{y}_{VEST}(t) &= C_{VEST}\mathbf{x}_{VEST}(t) + D_{VEST}\mathbf{u}(t), \quad t \in \mathbb{Z}. \end{aligned} \tag{3.59}$$

3.3 Complete MPC model: mechanical and vestibular systems series

The vestibular model (3.59) is a key feature for the MPC algorithm, but has to be integrated with a *mechanical model* of the platform. As proposed by Dagdelen et al. (2009), it is reasonable to consider as a complete model for the procedure the *series* between the mechanical and vestibular model. The linear accelerations and angular velocities of the real vehicle chassis are computed by a simulation software and then provided to the platform motion controller; the simulator is then moved accordingly, and that is the motion perceived by the on-board driver. A schematic is reported in figure 3.11. That being, the mechanical system acts like a “pre-filter” for the signals that will then be perceived by the pilot: in the ideal case, a precise model of the specific platform in use

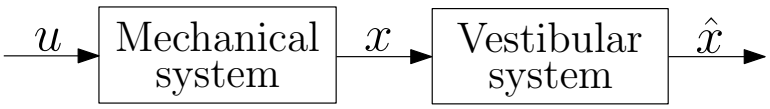


Figure 3.11: Vestibular and mechanical models cascade.

should be integrated with the vestibular model.

The present dissertation aims to an higher level of abstraction, hence the mechanical model cannot be so specific. D'Ambrosio (2010) proposed a generic second-order, LTI system as mechanical model for each DOF; with this approach, it is still necessary to set a proper value for the damping factor ξ and the natural frequency ω_n, increasing the tuning overhead (it may not be that immediate to derive such parameters); more, it is better to keep the model dimension as low as possible, to ease the task of the solver. For these reasons, together with the robustness of the MPC framework to (not too large) errors in the model, a one-step-delay model has been chosen as mechanical description for each DOF, with matrices

$$A_{m,a_i} = A_{m,\omega_i} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad B_{m,a_i} = B_{m,\omega_i} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{3.60a}$$

$$C_{m,a_i} = C_{m,\omega_i} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \qquad D_{m,a_i} = D_{m,\omega_i} = 0, \qquad i = x, y, z. \tag{3.60b}$$

The complete model

$$\Sigma_m = \{A_m, B_m, C_m, D_m\} \tag{3.61}$$

has as state $\mathbf{x}_m$ the direct sum of each of the states of systems in (3.60), so $\mathbf{x}_m \in \mathbb{R}^{12}$: the inputs are the same of system (3.59), and the outputs are the inputs delayed by one sample step.

3.3.1 Series system

The series connection of (3.61) and (3.59) yields to the system Σ with matrices

$$A = \left[\begin{array}{c|c} A_m & 0 \\ \hline B_{VEST}C_m & A_{VEST} \end{array}\right] \qquad B = \left[\begin{array}{c} B_m \\ \hline B_{VEST}D_m \end{array}\right] \tag{3.62a}$$

$$C = \left[\begin{array}{c|c} D_{VEST}C_m & C_{VEST} \end{array}\right] \qquad D = \left[\begin{array}{c|c} D_{VEST} & D_m \end{array}\right]. \tag{3.62b}$$

The resulting discrete system has *thirty-three* state variables, making the real-time MPC application a challenging task: this has to be taken in consideration in choosing the QP solver and in the implementation phase.

3.4 Optimization strategy

In section 3.1.5, it has be shown how the considered MPC problem can be expressed as a Quadratic Programming problem with constraints. The presence of the latter ones makes impossible to find an analytical resolution of the problem, hence the use of appropriate solvers is necessary: from the implementation point of view, the Quadratic Programming solver becomes the core of the MPC algorithm. In the application at hand, there are strict real-time requirements, since fast dynamics (control frequency of 100 Hz) call for small computation times: a generic QP solver is too limited so specialized algorithms that exploit the particular structure of QPs arising in MPC are needed. These can be divided in two main categories, *offline* and *online* solvers.

The principle of offline optimizers is to *precompute* a solution for all the possible instances arising in the problem. In this field, *Explicit MPC* is a widely used technique. First proposed by Bemporad et al. (2002), the optimization step is computed offline before applying the control algorithm to the process. The rationale is to use the constraints to partition the state space in *polyhedral critical regions* in each of which the optimal control law is an *affine function* of the state. In this approach, only the critical regions and the parameters of the corresponding affine functions need to be stored, while the online computational effort is limited to the evaluation of the region which the current state belongs to and its correspondent affine function. The drawback of the method is that it can only been applied to low-dimensional problems, since the offline computation burden has exponential growth in the size of the problem and the number of regions increases with the number of constraints, making the evaluation slower. Moreover, the offline computation makes the online tuning very difficult, and this is an important disadvantage for the specific application, since the presence of the human driver in the loop makes the tuning phase crucial. Pannocchia, Rawlings, and Wright (2007) propose a suboptimal solution to reduce the problem complexity, but in the same work it is remarked that in practice, this offline approach is useful for SISO problems with small N_p, but as the problem dimensions increase in terms of states, inputs, outputs and prediction horizon, other approaches should be preferred.

On the basis of such considerations and given the characteristics of the MC problem, the use of *online QP solvers* has been preferred in the specific application at hand. Online, quadratic optimization for MPC can be successfully addressed by using algorithms belonging to two main categories, namely

1. *Interior Point* (IP): these methods exploit the convexity of the cost function and have polynomial complexity (Wright (1997)). IP methods are suitable for dealing

with linear and quadratic programming problems and the bound on the complexity is an ideal feature in the MPC case, however, they lack effective *hot-start* strategies, i.e. the capability of exploiting the results from the QP solved at the previous step to obtain a "smart" starting point for the current QP, hence reducing the computational effort. Studies on this topic have been carried on (e.g. see Wang and Boyd (2010) and Shahzad and Goulart (2011)), but a general, efficient solution has not yet emerged;

2. *Active Set* (AS): inspired by the same idea at the base of the explicit approach, and expressible both in *primal* (Gill, Gould, Murray, Saunders, and Wright (1984a); Gill, Murray, Saunders, and Wright (1984b)) and *dual* (Goldfarb and Idnani (1983); Bartlett and Biegler (2006)) form, these methods aim at finding the current critical region without a pre-computation phase. Inside that region, the QP solution depends affinely on the state of the problem. Despite the fact that polynomial complexity is not always achieved (the worst case is exponential in the size of the problem, as proved by Klee and Minty (1972)), AS methods provide a hot start strategy based on the hypothesis that the active set does not change much from two subsequent QPs during the MPC process. By moving along directions derived from the parameters from the old problem to the new one, exploiting the convexity of the parameter space and considering only a subset of constraints to reduce the complexity of the problem, intermediate QPs are solved, until the solution of the global problem is found. In case of boundary crossing of the current set, the new active set is calculated online.

After having analyzed different solutions, an Active Set method has been chosen to deal with the MPC problem described in this dissertation.

3.4.1 Online Active Set strategy: `qpOASES`

Going a bit more into the details of classic AS strategies, the idea is that at time t of the MPC procedure, there exists an optimal solution $\Delta\mathbf{U}^*$ minimizing (3.23), to which correspond specific values of vector c in (3.22) and constraints vector b in (3.27), both dependent on the state $\mathbf{x}(t)$. Note that starting from an optimum point the feasibility at the start of the iterations is assured. At the next time step, the "initial state" for the problem is $\mathbf{x}(t+1)$, the corresponding problem parameters $c(\mathbf{x}(t+1))$ and $b(\mathbf{x}(t+1))$ are calculated and the new solution $\Delta\mathbf{U}^{*,new}$ has to be computed. At this point, to reduce the complexity of the problem, only the set of constraints that are *active*, i.e. satisfied by equalities are considered, and the corresponding QP problem is solved. The feasibility

of this intermediate solution for the general problem has to be verified, as well as its optimality; in case one of this two conditions doesn't hold, new constraints are added or removed, and the process iterated until an optimal solution is found. A large body of literature on this method (derived from simplex) can be found, e.g. in the works by Nocedal and Wright (2000) and Bartlett, Biegler, Backstrom, and Gopal (2002). However, it is clear from this simple sketch how the complexity can grow exponentially when a large number of constraints is present and only few (at worst one) are added or removed at each iteration.

The chosen AS algorithm has been proposed by Ferreau et al. (2008), and has features that exploit the structure of a QP problem derived from MPC. Here the main idea is to introduce a smart recalculation way that grants to move on a "feasible" line from $x(t)$ to $x(t+1)$, estimating the progress through the new overall optimum when upgrading the c and b vectors while keeping all the intermediate problems feasible: all the steps from the old optimal solution to the new one are carried on along *homotopic* variations of the parameters. KKT matrix and Lagrangian multipliers are implied in the procedure (see Boyd and Vandenberghe (2004) for details on these fundamental elements of optimization theory): at each iteration the homotopy has a maximum step length that can be calculated, once the maximum step length is reached the new optimum is obtained. This interpretation of the AS method is specifically thought for MPC purposes. In fact, in such kind of problems it is reasonable to assume that the system behaviour does not have dramatic changes during two successive time steps. Consequently, the corresponding solutions and active sets are close, thus motivating an homotopy-based approach, that can provide a noticeable improvement in the real-time performance.

The proposed method has a freeware `C++` implementation by Joachim Ferreau and the OPTEC group at University of Leuven, `qpOASES` (Optec, 2012), a ready-to-use package with real-time capabilities that offers some useful solutions for matching different real-time requirements, as the tunable limitation of the maximum number of active set recalculation per sample step and heuristics to assess the time required to complete the current optimization calculation, while limiting the possibilities of infeasibility if the procedure is stopped before reaching the optimum. This last feature is particularly interesting when hard real-time is requested, and is based on the idea that the iterating QPs are granted to be feasible by themselves, even it is not sure if they could be feasible, despite being sub-optimal, for the overall problem: by estimating the time that a change of active set requires, it is possible to evaluate if the problem could be solved in the sampling time or not, in this case the procedure is interrupted and a new search is started through a new homotopy to the new starting point (i.e. the new provided or estimated state of the

state space model) for the next time interval. If this new search is fast enough, previous computation can be continued to find the optimal solution at present time and apply the calculated input to the system, otherwise a recovery solution has to take place before the procedure could resume with the new calculated input. Major analysis and details can be found in Ferreau (2011).

3.5 Rationale and remarks

There are some important aspects of the MPC algorithm realized for the Motion Cueing problem. As confirmed by the implementation that will be illustrated in the next chapter, the majority of the driving simulator platforms come together with position controllers that regulate their dynamic behaviour. Typical industrial PLC with ad-hoc implementation, this devices have to be considered as black-box elements in the system: it is very rare the case where explicit information are provided. The task of a general-purpose Motion Cueing algorithm is then not to directly control the actuators, but to generate reliable trajectories to be passed to this position controller. Our algorithm has been thought with this aspect in mind, and model (3.62) has outputs and inputs coherent with this idea. The conceptual scheme of the MC procedure is summarized by the block scheme in figure 3.12, and is composed by the following steps:

1. obtain the current useful vehicle states d, i.e. translational acceleration and angular velocities calculated on the driver's eye-point, from the simulation software;

2. obtain the *perceived* acceleration r by filtering d via the vestibular system model, thus generating the reference signal for the MPC algorithm;

3. compute via MPC the displacement signal p to be passed to the platform control system to achieve the desired behavior on the eye-point.

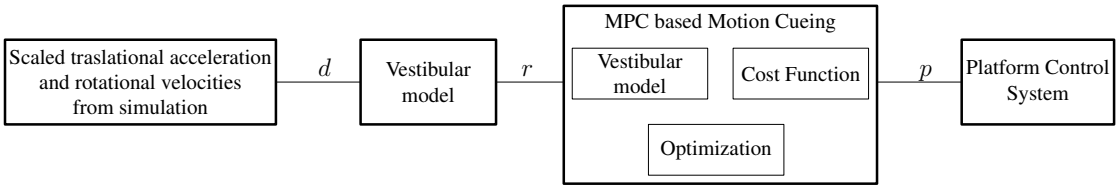


Figure 3.12: Proposed MPC algorithm for Motion Cueing.

The proposed model is LTI. This is surely an approximation of the reality, and it is trivial to deduce, even from the brief details given about its physiology, that the human

vestibular system and the interactions between its elements are most likely non-linear. Two main reasons lead to accept this approximation

1. nonlinear MPC is still a challenging task: research on the topic as been carried on over the years (Mayne and Michalska, 1990; De Nicolao, Magni, and Scattolini, 2000; Camacho and Bordons, 2007) and solution packages have been implemented, like `ACADO toolkit` (ACADO Toolkit, 2012), considering nonlinearities in the model, in the cost function and in the constraints. Nevertheless, this is still a great challenge when applied to practical problems, especially when hard real-time is required. In this sense, Quadratic Programming is a much more affordable task, and is by now preferable, letting the robustness of receding horizon approach compensate for imprecise models;

2. model identification of human organs is challenging, and the experiments by themselves can be tricky: for the vestibular system the complexity is increased by the difficulty to derive reliable measurements and estimations. These drawbacks justify the choice of simpler, linear models that can have a frequency behaviour compatible with the reality.

To assure the reliability, the choice of the model has been made in collaboration with a professional otolaryngologist.

As can be seen by analyzing the structure of (3.3), it is necessary to produce reference trajectories for each of the output variables. This can be done by using the *simulation environment*, where perceived transactional accelerations and angular rates are generated, and then scaled prior to be used in the MPC algorihm: in this sense, state estimation is not a problem since each physical quantity that may be of interest is directly available. As far as the design of the MPC algorithm is concerned, it would clearly be convenient to make use of the widest prediction/control window possible, if reliable information on the future driver's behavior is available. However, hard real-time constraints and the possible lack of capability to predict the driver's behavior may limit the length of the prediction/control window in practical situations. For these reasons, to design a general purpose Motion Cueing algorithm, it is preferable to first provide an effective algorithm where information about the future driver's behavior is not required and where a small prediction window is used. In the proposed solution, the MPC is designed to keep the reference trajectory constant along the prediction interval, and the length of the window N_p becomes a tunable parameter that can be varied to obtain the desired tracking performance. This is a common strategy in MPC-based tracking problems for system with slow dynamics (Maciejowski, 2002): this is not properly our case, but keeping N_p small

enough gives in any case good results, and the variations of that parameter becomes a really useful element in the overall tuning procedure.

A valuable MC algorithm must include a Washout Action to keep the platform within its operational limits. MPC provide an alternative to the classical, high-filtering procedure by imposing constant zero references for the position of the all six DOFs and for the velocities of the longitudinal ones are used. This is an effective realization of Washout, easy to handle and to modify on the fly: in particular, playing on soft and hard spatial constraints could lead to a platform exploitation impossible to reach with classical procedures.

By integrating the gravity effect inside the model as described by (3.50) and (3.54), the low-frequency tilt coordination correction is automatically achieved giving an important contribution to the tracking of the perceived simulation signals. This fact can be seen as one of the major advantages of the MPC approach to the design of the motion cueing algorithm, namely, a tilt coordination correction can be obtained as the result of a model based optimization procedure and not a simple coupling of filters, whose frequency behaviour could be harder to integrate. In such conditions the low-frequency component of the desired perception signal is covered by tilt coordination and the high-frequency component is reproduced by using the translational DOFs. Note that it is possible to add constraints in the cost function to limit the rate of the tilt coordination and hence to limit in a simple way the undesired rotation that occur when tilting.

Beside the already explained role of N_p, the weights in (3.3) become the *tuning parameters* of the MC algorithm, and are manipulated to achieve a satisfactory reproduction of the perceived accelerations and velocities in the simulated vehicle. Here the key fact is that the user has a direct knowledge of which physical quantities will be affected by the variation of a certain parameter, hence the overall procedure becomes way more fast and reliable than the classical one. Moreover, acting on the weights of the cost function in the tuning phase allows to modify a specific behaviour of the platform while letting other characteristics barely unaffected (at least in presence of little variations). As an example, consider the case in which one wants to limit the longitudinal displacement while keeping unchanged the tilt coordination: an increase in the position weight will cause lower displacements with almost the same pitch usage and a tracking of acceleration very similar to the original. Tilt coordination is another feature that can be easily regulated by acting on weights. Practical examples will be given in the next chapter. The latter coefficients that complete the tuning capabilities are the scaling factors, useful to allow an effective usage of the platform working area.

The problem formulation is robust and rarely leads to infeasibility (as proven by many practical tests). In any case, a back-up strategy is provided, where at each step the first

samples of the calculated optimal sequence are stored in a buffer and used in case of infeasibility of the subsequent problems: in case of repeated infeasibility, the platform is safely stopped.

4

Application: VI-DRIVESIM

The Model Predictive Control-based Motion Cueing algorithm has been desgned keeping in mind the practical implementation. This has been made possible by the collaboration with VI-GRADE S.R.L.(VI-Grade, 2012), renowned international company with a solid and long-lasting experience in the field of software dynamic simulation. VI-GRADE is one of the promoters of the entire project, and made available an innovative dynamic simulator together with the expertise in vehicle simulation. The challenge was particularly interesting, since the device has a pretty new concept and the basic MC algorithm it was provided with showed major limitations. The MPC procedure previously exposed has been fully implemented in high-level programming language (C++) and interfaced via a GUI to tune online the parameters.

We will give now a in-depth presentation of the device, and some expedients adopted to ease the implementation, taking advantage of the characteristics of the platform. Results from test carried on by a professional driver on the testbed will be provided and

analyzed, as well as simulation studies with the combination of effective reference signals for exploit better the predictive capabilities. Finally, the possibility of and hardware, FPGA-based implementation of the procedure will be discussed.

4.1 The VI-DRIVESIM platform

VI-DRIVESIM is a set of modern tools made available from VI-GRADE with the aim of providing automotive OEMs with the instruments to simulate at best new products (both brand-new vehicles and innovative components for the existing ones) without the need of realizing a prototype and test it in a real track. This idea greatly improves flexibility in the design phase, exploiting the potential of the *Hardware-In-the-Loop* paradigm (Raman, Sivashankar, Milam, Stuart, and Nabi, 1999; Sung Chul, 2005; Taehun, Jihoon, Kihong, Jeongho, Kyu, Kangwon, Soo-Jin, and Young-Jun, 2006) and reducing the expenses. The package consists of softwares for simulate the vehicle dynamics physical engine as VI-CARREALTIME, which provides a highly reliable representation of the real vehicle behaviour, and driving simulators, one of which is *dynamic*. In figure 4.1 the platform used as testbed in this dissertation is shown. Its peculiarity is in the mechanical structure: by using *electric linear actuators* instead of the classic, hydraulic hexapodal structure, it is able to achieve satisfactory results in physical simulation with a relatively small size hardware (4m long, 5.5m wide, 3m high, considering maximum displacements), that can fit standard laboratories environments whereas it has been seen that traditional, large dimensional, hexapodal platform can require dedicated hangars for high-level performances: table 4.1 reports a qualitative comparison with traditional devices.

	VI-DRIVESIM	Equivalent hexapod
Base area	1	2
Volume	1	4
Power consumption	1	2
Stroke	1	1
Rate	1	1
Weight	1	3

Table 4.1: Qualitative comparison between the testbed platform and a classic hexapod.

The architecture is based on an upper plate, where the car's cockpit and seat is installed, able to move in vertical direction and rotate along pitch and roll angles, linked to the base of the device, which can move along longitudinal and lateral directions and perform yaw rotations. A deeper observation of figure 4.1 can help the comprehension of the basic structure. It is easy to understand that the base moves along three completely *decoupled*

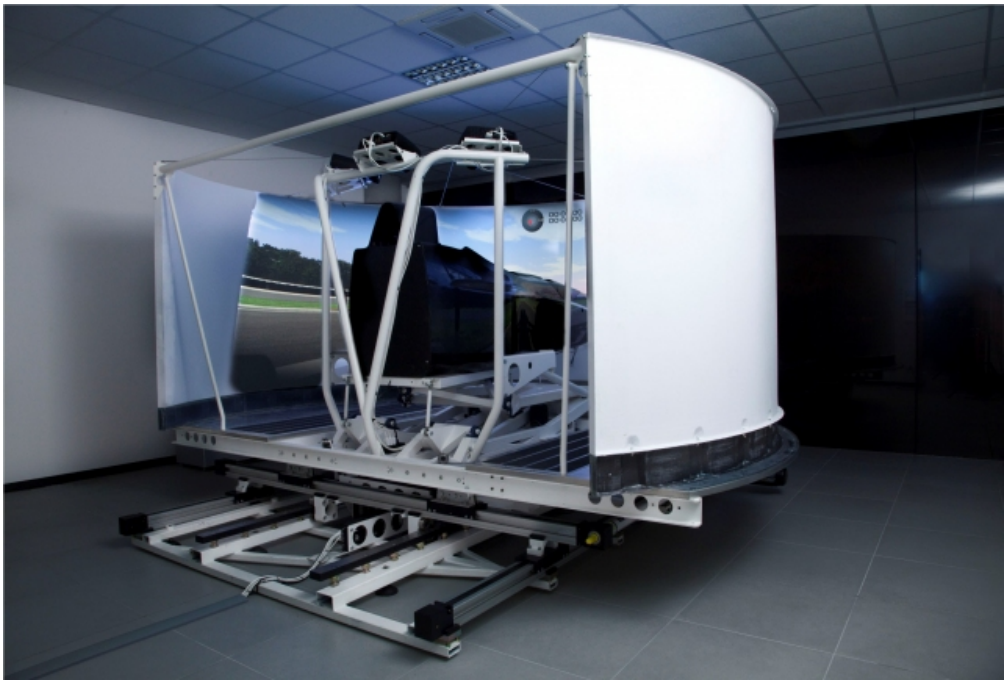


Figure 4.1: VI-DRIVESIM dynamic platform.

degrees of freedom, while the three DOFs of the upper plate are *partially coupled*. The screen covers more than 180 deg and moves in agreement with the platform to guarantee full immersion of the driver in the virtual environment, this aspect is detailed in figure 4.2. Finally, force feedback on the steering wheel and the braking system enhances the "driver's feeling" of the vehicle behaviour.

The platform dynamic performance reported in table 4.2 highlights the limitations of the operational space, with maximal linear excursions of 1 m. This fact makes the role of the MC algorithm crucial. Note how the three DOFs related to the base, and yaw more than the other, have much wider working area. For what concerns the coordinate system, x, y and z correspond to longitudinal, lateral and vertical directions; pitch θ is a rotation around y axis, roll ϕ is around x and yaw ψ is around z. The MC strategy has to provide the displacement references to the control system of the platform, which is assumed to be able to perfectly track the reference signals, with a fixed time delay. In the specific case, the position controller is an industrial PLC, GEO BRICK from DELTA TAU LTD (Delta Tau Data Systems, 2012). It is a state-of-the-art controller for this kind of applications, and is used in the field of simulators also for aerospace purposes. From our point of view, this is a black-box controller, whose operating logic is unknown. The MPC algorithm will provide the input for this component, whose task is to position its *end-point* (the spatial

Figure 4.2: VI-DRIVESIM: driver positioning.

Range	Position	Velocity	Acceleration
x	1m	1.3m/s	3.3m/s^2
y	1m	1.3m/s	3.6m/s^2
z	0.5m	0.9m/s	4.9m/s^2
Roll	30deg	112deg/s	600deg/s^2
Pitch	24deg	61deg/s	600deg/s^2
Yaw	50deg	61deg/s	240deg/s^2

Table 4.2: Platform performance.

point where the control is actuated) to the desired location. The end point is, in this case, the *eye-point*, i.e. the position of a virtual "vestibular" sensor located in the center of the head: the aim is clear, as if the control acted directly on the vestibular system so that the produced perceived signals are as close as possible to the ones actually sensed.

4.2 Implementation

The test simulator requires a control frequency of 100 Hz, commensurate with the involved dynamics. This is a challenging task indeed, from the implementation point of view. Model (3.62) has a remarkable size by itself, and with a control horizon N_c it becomes a QP problem of kind (3.23) and size $N_c \times n_{in}$; the correspondent number of constraints is, from (3.27),

$$(2 \times N_c \times n_{in}) + (2 \times N_c \times n_{in}) + (2 \times N_p \times n_{out})$$

hence the real time request is non-trivial. First of all, the model is reduced to *four sub-models*, corresponding to four QP problems of smaller dimensions. In this way, parallelizing the calculations (e.g. using different synchronized PCs) the overall elapsed time is significantly reduced. As seen in the previous section, the architecture of the considered platform is such that three of the DOFs are naturally decoupled. For what concerns the z direction, we have that at its maximum excursion it is strongly coupled to pitch θ and roll ϕ: the last 20 cm are obtained by extending together the actuators responsible for those angular motions. But if we set a constraints on the maximum vertical position to 0.3 m, we get that the motion is handled by the vertical actuator only, and this DOF becomes decoupled. It is intuitive to understand that the vertical direction is less incisive in a car-oriented dynamic simulator with respect to the other two linear DOFs, and as results from table 4.2 it would be in any case 1/3 of x and y maximum displacement, an admissible approximation. On the other hand, this is coherent with the linear approximation (3.41) where the absence of angular influence on a_z is evident. Nevertheless, it is necessary to consider a coupling between x and θ on one hand, y and ϕ on the other to take into account the tilt coordination (evident from (3.41)): the first one is of particular relevance in the four-wheels framework, hence it will now be given a deeper insight of the realization of this subsystem, that we call $\Sigma_{x\theta}$. The construction of the model is similar to the one already described in the previous chapter. First of all, the mechanical retard is considered the same in all the DOFs without interaction, as in equation (3.61): from the model for the single DOF given in (3.60) we obtain

$$\Sigma_{m,x\theta} = \{A_{m,x\theta}, B_{m,x\theta}, C_{m,x\theta}, D_{m,x\theta}\} \tag{4.1}$$

where

$$A_{m,x\theta} = \left[\begin{array}{c|c} A_{m,a_x} & 0 \\ \hline 0 & A_{m,\omega_y} \end{array}\right] \qquad B_{m,x\theta} = \left[\begin{array}{c|c} B_{m,a_x} & 0 \\ \hline 0 & B_{m,\omega_y} \end{array}\right] \tag{4.2a}$$

$$C_{m,x\theta} = \left[\begin{array}{c|c} C_{m,a_x} & 0 \\ \hline 0 & C_{m,\omega_y} \end{array}\right] \qquad D_{m,x\theta} = 0 \tag{4.2b}$$

For what concerns the vestibular model, the procedure is similar and recalls what has been previously presented. The model for the involved semicircular canals is (3.44) for $i = y$ (pitch corresponds to a rotation around the lateral axis), for the otoliths in the longitudinal direction tilt coordination must be included. Following the reasoning in equations (3.48)–(3.49)–(3.50)–(3.51) relative to only the two Degrees of Freedom of

interest, we have

$$\mathbf{a}_x = H_x \begin{bmatrix} a_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} 1 & g \end{bmatrix} \begin{bmatrix} a_x \\ \theta \end{bmatrix}. \tag{4.3}$$

By augmenting the state as $\mathbf{x}_{\overline{OTO},x\theta} \triangleq [\mathbf{x}_{OTO,x}^T \ \theta]^T$ and setting $\mathbf{u}_{\overline{OTO},x\theta} \triangleq [a_x \ \dot{\theta}]^T$, the system matrices are

$$A_{\overline{OTO},x\theta} = \left[\begin{array}{c|c} A_{OTO,x} & g \\ \hline 0 & 0 \end{array}\right] \qquad B_{\overline{OTO},x\theta} = \left[\begin{array}{c|c} B_{OTO,x} & 0 \\ \hline 0 & 1 \end{array}\right] \tag{4.4a}$$

$$C_{\overline{OTO},x\theta} = \left[\begin{array}{c|c} C_{OTO,x} & 0 \end{array}\right] \qquad D_{\overline{OTO},x\theta} = \left[\begin{array}{c|c} D_{OTO,x} & 0 \end{array}\right]. \tag{4.4b}$$

The output here is the perceived longitudinal acceleration with the tilt contribution included, $\mathbf{y}_{\overline{OTO},x\theta} = \hat{a}_x$. As before, integration is necessary to explicitate velocities and positions and is obtained via the integral system

$$A_{I,x\theta} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B_{I,x\theta} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}. \tag{4.5}$$

Combining all together, the vestibular model related to the longitudinal direction is

$$A_{V,x\theta} = \begin{bmatrix} A_{SCC,y} & 0 & 0 \\ 0 & A_{\overline{OTO},x\theta} & 0 \\ 0 & 0 & A_{I,x\theta} \end{bmatrix} \qquad B_{V,x\theta} = \begin{bmatrix} 0 & B_{SCC,y} \\ B_{\overline{OTO},x\theta} & \\ & B_{I,x\theta} \end{bmatrix} \tag{4.6a}$$

$$C_{V,x\theta} = \begin{bmatrix} C_{SCC,y} & 0 & 0 \\ 0 & C_{\overline{OTO},x\theta} & 0 \\ 0 & 0 & I \\ & 0 & \end{bmatrix} \qquad D_{V,x\theta} = \begin{bmatrix} 0 & D_{SCC,y} \\ D_{\overline{OTO},x\theta} & \\ 0 & \\ 0 & 1 \end{bmatrix} \tag{4.6b}$$

with state, input and output

$$\mathbf{x}_{V,x\theta} = \begin{bmatrix} \mathbf{x}_{SCC,y}^T & \mathbf{x}_{\overline{OTO},x\theta}^T & p_x & v_x \end{bmatrix}^T \tag{4.7}$$

$$\mathbf{u}_{V,x\theta} = \begin{bmatrix} a_x & \dot{\theta} \end{bmatrix}^T \tag{4.8}$$

$$\mathbf{y}_{V,x\theta} = \begin{bmatrix} \hat{\omega}_y & \hat{a}_x & \theta & p_x & v_x & \dot{\theta} \end{bmatrix}^T. \tag{4.9}$$

This latter model is then discretized to system $\Sigma_{VEST,x\theta}$ and combined in series with (4.2) to obtain the system $\Sigma_{x\theta}$

$$A_{x\theta} = \left[\begin{array}{c|c} A_{m,x\theta} & 0 \\ \hline B_{VEST,x\theta}C_{m,x\theta} & A_{VEST,x\theta} \end{array}\right] \qquad B_{x\theta} = \left[\begin{array}{c} B_{m,x\theta} \\ \hline B_{VEST,x\theta}D_{m,x\theta} \end{array}\right] \tag{4.10a}$$

$$C_{x\theta} = \left[\begin{array}{c|c} D_{VEST,x\theta}C_{m,x\theta} & C_{VEST,x\theta} \end{array}\right] \qquad D_{x\theta} = \left[\begin{array}{c|c} D_{VEST,x\theta} & D_{m,x\theta} \end{array}\right]. \tag{4.10b}$$

With the same procedure, system $\Sigma_{y\phi}$ can be obtained, while Σ_z and Σ_ψ are straightforward. The two coupled systems have size 11, the vertical one has size 6 and the yaw one 5 (the distinction between the latter is due to the number of integrators, z DOF has to be integrated twice to get the position while ψ only once). The four consequent subproblems have much smaller dimensions and allow for faster computation. Combined with efficient multi-threading parallel solutions the computational burden is improved with respect to the complete system, and the decoupling approximation becomes an excellent compromise to get fast real-time performance.

For what concerns the realisation of the software, there has been a first implementation with MATLAB to validate the procedure and to have a useful tool for simulation purposes, which has then evolved to a more efficient and complete C++ implementation, but the computational core, i.e. the QP solver qpOASES, has always been compiled from C++ sources, available online (Optec, 2012). In this way the critical step of the algorithm has always worked at its best in term of computational time. Nevertheless the solver has to be tuned as well as the MC parameters. In particular, the fundamental features to be regulated are

1. the *hot-start* strategy. As stated, this particular solver has been chosen among others because it is based on an active set strategy, which is capable of exploiting the results of the previous optimization step to speed up the current step. Despite the fact that disabling this feature would ease the tuning setup (solving a completely new problem at each step would ideally make possible instantaneous change in the weights and constraints), the fast system dynamics involved require it to be enabled;

2. the *maximum number of set recalculations*. In subsection 3.4.1 it has been shown how the idea at the base of the optimizer allows to move from the previous solution to the new one via a path in the feasible space along which all the QPs are granted to be feasible; each new QP requires the calculation of a new active set. This, in the worst case, can be done an exponential number of times in the size of the problem, so in the package it is possible to set explicitly the maximum number

of recalculations allowed. This enables an heuristic that try to force the problem to be solved respecting this parameter, possibly at the expenses of single iteration time. Setting a reasonably small value for this number has proved to improve the calculation time, with respect to the default, "high" value;

3. the *maximum allowed CPU time*. This is an interesting feature of `qpOASES`: from the analysis of the previous iteration, the solver is able to derive an estimation of the most likely time requested for a single, intermediate iteration (i.e. set recalculation) and, together with an heuristic which estimate how "near" is the algorithm to the correct active set, it is able to evaluate if the optimum calculation will require more time than it is specified. The value has been set to a safety value that allows for the recovery actions to take place.

The last two points are critical: together with the case of infeasibility of the problem, these are the situations where the solver fails with consequent unsafe conditions. These are handled according to the following rationale

1. at each iteration, even if only the first optimal input is applied, the first part of the sequence is saved in a buffer. If the solver fails, the other predicted and saved inputs are fed both to the platform and as starting point for a new QP problem, letting the algorithm iterate to regain feasibility;

2. if the solver is not able to calculate a feasible solution, the platform is safely taken back to the neutral position while a new QP problem is initialized from scratch.

This procedure has proved to be an affordable way of handling dangerous conditions.

The implementation of the tuning phase follows the same principle. The Motion Cueing procedure is tuned by varying parameters as weights, the constraints, the control and prediction horizons: all elements that modify the structure of the related QP problem. First of all, as already highlighted, the reference in the prediction horizon is held constant and equal to the current value: this simplifies the problem and imposes an initial phase to find the correct size in order to have a satisfying tracking action as well as affordable computation time (this parameter determines the dimension of the QP), while assuring that the problem remains feasible. Moreover, the control horizon is set equal to the prediction one, so that we have one less parameter to regulate: it would be easy, if needed, to introduce this feature in the future. To overcome the criticalities of on-the-fly modification of the structure, the procedure is realized in order not to directly apply the new parameters to the device. Instead, when all the regulations are done and confirmed, the platform is carried back to its neutral position while the parameters update, a new

QP starts and regains full operativity. At this point, the control is applied again to the platform in a safe way by varying a scale factor.

All this procedure would be very complicated if managed only by "console" commands, so the realisation of a Graphical User Interface was the logical subsequent step. Once again the first prototype has been realized in MATLAB to validate the procedure but the final aim of producing a tool as close as possible to a professional one pushed to a full implementation in high-level programming language. Having the core solver already implemented in C++ and the great real-time capabilities of this language made its choice obvious, so all the algorithm has been converted, and a new GUI produced.

To take advantage of the parallel structure of the problem, the implementation runs on a LINUX-based operative system with features optimized for the parallel calculus provided by CONCURRENT (Concurrent Real-Time, 2012). The interface with the platform is achieved by tools from ANSIBLE MOTION (Ansible Motion, 2012).

4.3 Results

Experimental results obtained during a professional driver training session on the platform are reported in this section. The simulated vehicle is a "hot hatch"-class car, namely a Volkswagen Golf R "digitally" boosted to have a maximum power of 350 HP. The virtual test track is a digital version of the Calabogie track in Canada; figure 4.3 shows a sketch of the track. As previously stated, the test platform is almost decoupled for all the 6 DOFs, so we analyze the dynamics of the subsystems separately. As an example of coupling effects, particular attention will be given to longitudinal-pitch one, hence examining the quality of longitudinal acceleration, that has to be reproduced as faithfully as possible (in terms of driver sensations) by operating the platform with longitudinal and pitch motions. The behaviour of the other subsystems will be shown as well, to give an idea of the whole platform exploitation. Insights on of the effect of the tuning procedure will be given, disclosing the effects of parameters variations. A simulation example will be shown to verify the improvement achievable by applying the "look-ahead" (i.e. prediction) technique.

4.3.1 MPC-based vs. classical Motion Cueing: motion inversion

Beside the tuning difficulties, one of the major problems for classic MC is the calibration of the Washout Action. Having no possibility to specify a priori the physical limitations of the device, the only available choice is to introduce a conservative action to keep the platform as near as possible to the center, to assure safe conditions. This is traditionally achieved

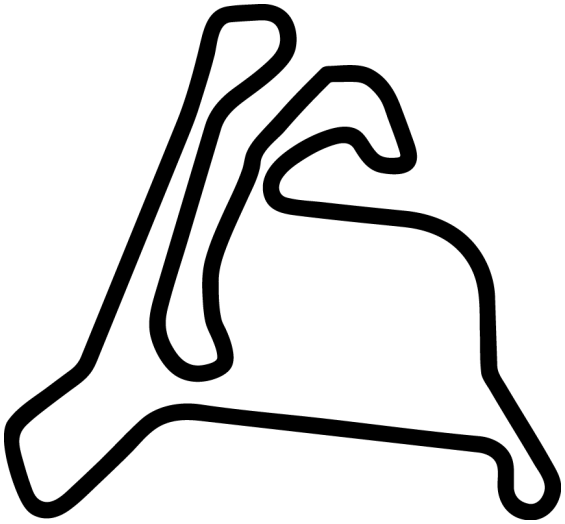


Figure 4.3: Calabogie track.

by a high pass filter set *after* double integration (i.e. on positions). Beside the fact that in most cases the platform is poorly exploited, this filtering is likely to determine motion sickness: by eliminating a slot of low frequencies, it may happen that the continuous component of accelerations signals, even in presence of tilt coordination contribution (traditionally set at low values as stated in Chapter 2), become close enough to zero to determine an inversion of the sign of the actual acceleration with respect to the visual cue. This is clearly a major problem of the classic approach. Model Predictive Control greatly improves this situation by the introduction of constraints on position, that make explicit the limitations the platform is subject to. In this way the reference trajectories are generated by an algorithm that explicitly knows the current position of the eye point and how far it is from the limits, and automatically determines the appropriate behaviour to preserve both safety and perception reliability. More, it is possible to enable a simple constraint directly on the sign of the acceleration, to double-check the coherence of the motion. The effect of this feature is evident in figure 4.4, where the signal from a classic algorithm is compared to the one produced by the MPC approach. The improvement is evident, and the noticeable motion inversion is avoided with the optimal control algorithm.

4.3.2 MPC-based Motion Cueing: longitudinal - pitch coupling

Let's now see in details the performance along the longitudinal direction. The motion cueing algorithm is set-up so that the platform working area in the longitudinal direction is exploited at best, while avoiding motion sickness due to the tilt coordination correction.

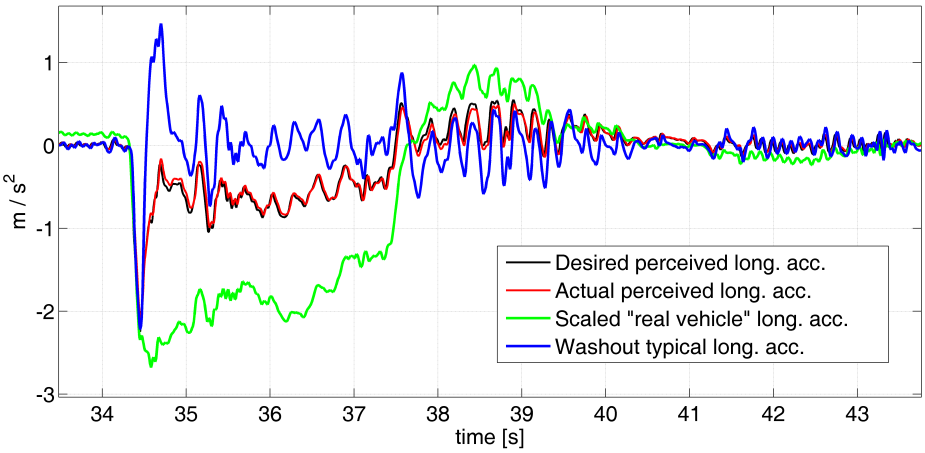


Figure 4.4: Effect of MPC on motion inversion.

This latter statement cannot be proved by means of figures or measured data, nevertheless the feedbacks from the professional drivers during practical tests have always been satisfactory in this sense.

The specific values of the tuning parameters (weights) are not of primary importance, but it is useful to specify the value of N_p, that is $N_p = 30$: at a control frequency of 100 Hz, it means a prediction horizon of 0.3 seconds. We recall that in the practical implementation the future reference is kept constant in the prediction window.

In figures 4.5 and 4.6, tracking of the perceived accelerations and angular velocities is illustrated, and the correspondent longitudinal and angular platform displacement are shown in figures 4.7 and 4.8, respectively. It can be seen that the platform actually exploits all of its operational area. It is worth remarking again that this is achieved *w*ithout any filtering action, as would be required in a classical, washout filters based approach. Also, the perceived acceleration is tracked almost perfectly (figure 4.5) with a determining contribution from *tilt coordination*. Focusing on the pitch displacement, tilt coordination correction can clearly be seen and has a peak (absolute) value of 0.08 radians (ca. 5 degrees): this cannot be related to the suspensions dynamics, since the simulated car has stiff components (comparable to a sport car). In this respect, it is of interest a frequency analysis of pitch and longitudinal contributions to the global perception in the platform. The comparison between the FFT of these two signals, reported in figure 4.9, shows that the pitch action is responsible of the low frequency contributions only. The exploitation of tilt coordination is another crucial feature of this MPC approach to Motion Cueing. As of now, tilt coordination has been considered “a necessary evil” to reproduce low frequency accelerations: basically obtained by a low-pass filtering action, the coupling with other

motions has always been critical and very likely a cause of motion sickness, to the point that drivers often ask for it to be disabled. The advantage of a safe introduction of this feature is therefore evident: more, the presence of small-values *perception thresholds* (the value of rotational velocities under which the motion is no more detectable by the semicircular canals, and the same principle holds for the otoliths) contribute to set smart constraints to improve the overall feeling.

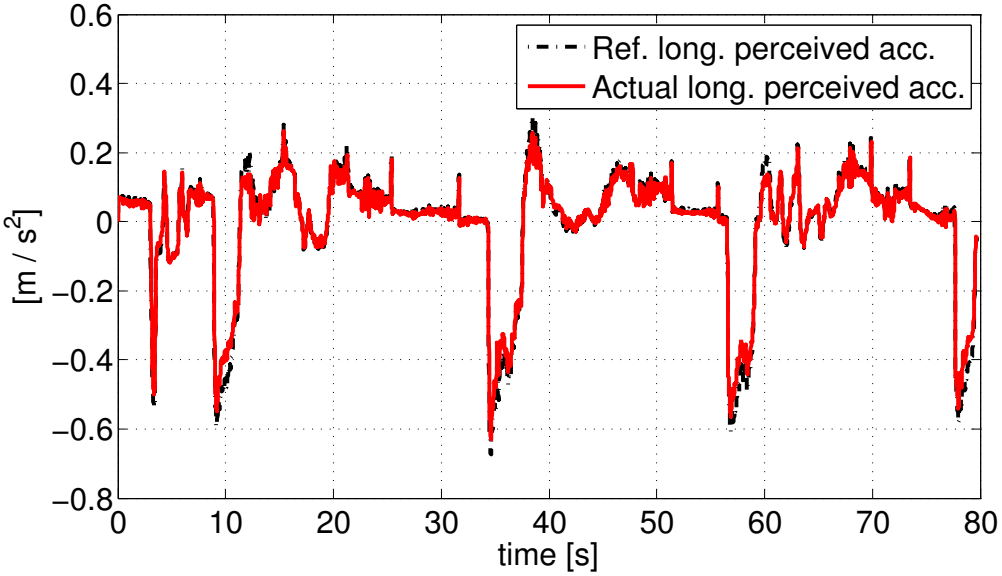


Figure 4.5: Perceived longitudinal acceleration tracking with the best tuning configuration.

The tuning procedure followed to achieve such results is very intuitive and it is based on the "natural interpretation" of the weights in the cost function. As stated, the crucial step of the tuning is handled mainly via manipulation of the weights in the cost function. As an example, let's suppose that it is requested to limit the displacement in longitudinal direction. To this aim, the weights values in x position are increased by 20%: to keep the cost function at a value as small as possible, the displacement is consequently reduced, but the optimality of the solution yields only to minimal losses in tracking performance for the perceived longitudinal acceleration (figures 4.11, 4.10). It is worth noticing that such performance degradation doesn't come at the expenses of "perception quality" (the produced signal is quite close to the reference) and does not introduce any discernible acceleration inversion, as it would typically be the case in washout-based MC, due to the presence of high pass filters.

Another interesting case is when it is requested to limit the effect of tilt coordination. The request is not uncommon, many drivers are sensitive to this feature and prefer to have it excluded to prevent simulator sickness. The procedure is analogous to the

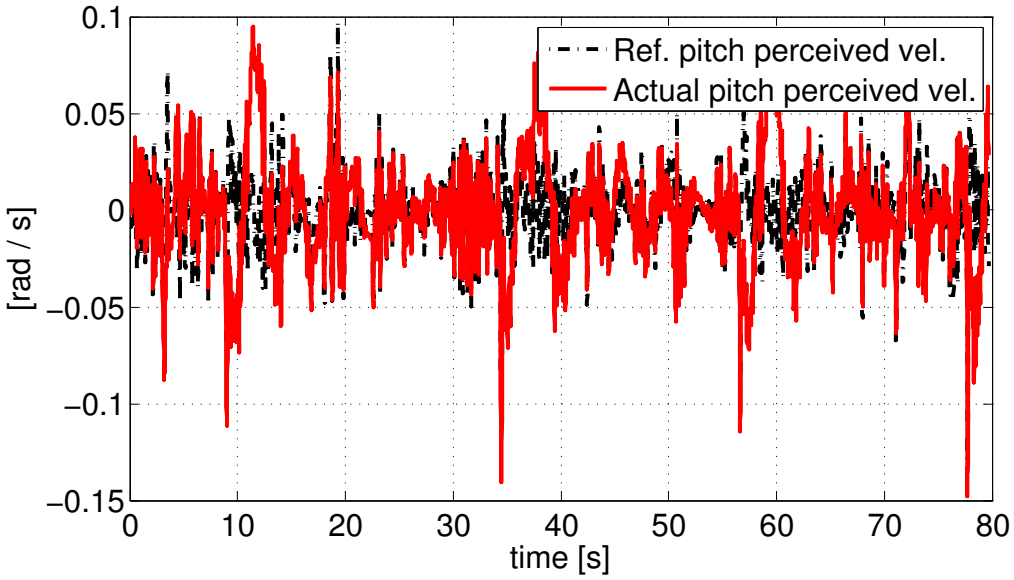


Figure 4.6: Perceived pitch velocity tracking with the best tuning configuration.

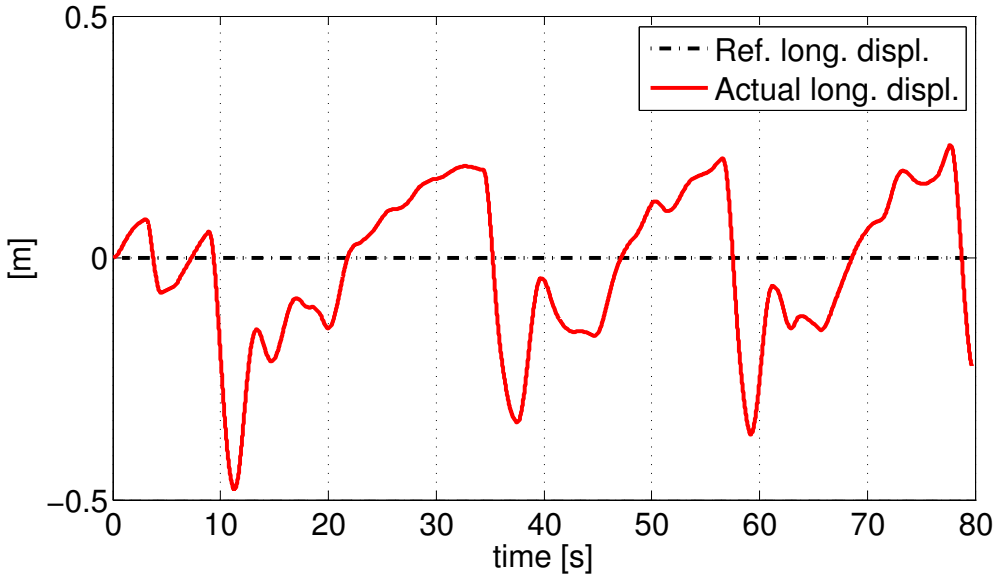


Figure 4.7: Longitudinal position displacement with the best tuning configuration.

previous stated one, and the weight on pitch position is varied (+20%). This case needs a further regulation: to limit the tilting effect means a greater demand for longitudinal displacement, to help compensating the loss of low frequencies. So, the scaling factors of the vehicle accelerations are increased, otherwise there would be the risk to saturate the longitudinal actuator and make the problem infeasible. Results are in figure 4.12: the tracking is still satisfactory and the pitch usage reduced. Note that the longitudinal

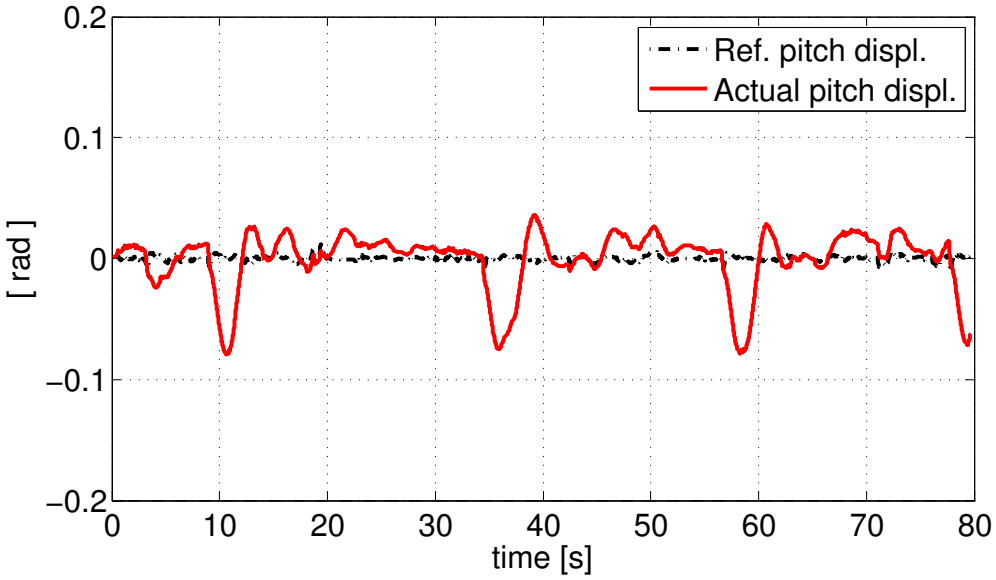


Figure 4.8: Pitch displacement with the best tuning configuration.

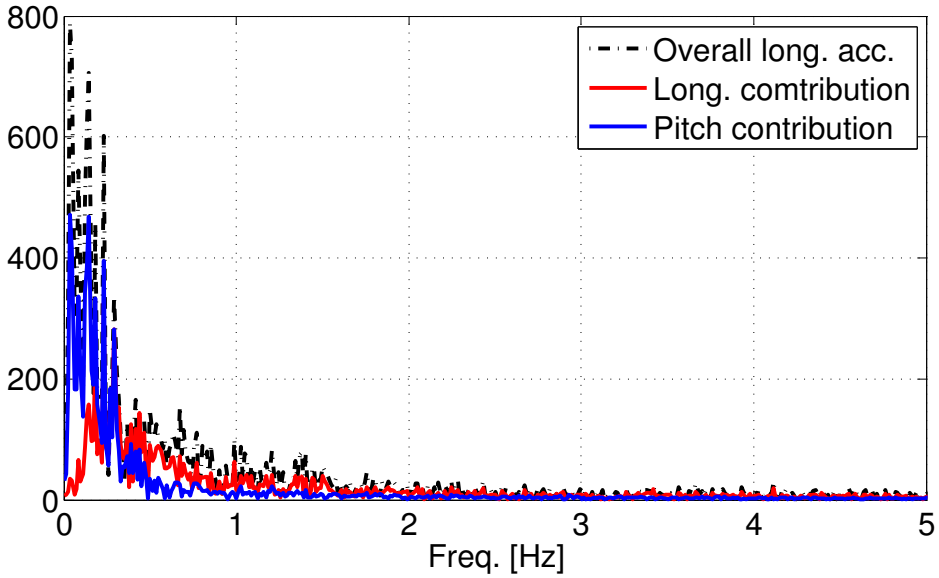


Figure 4.9: Longitudinal displacement and pitch frequency contribution to longitudinal acceleration: tilt coordination takes effect on low frequencies.

displacement is comparable to the one coming from the setup with tilt coordination active, which have lower scaling factor: in proportion, the overall platform exploitation favours the linear displacement.

It is of particular interest to study how the variation of prediction horizon length affects performance. By considering constant reference values for outputs along this time

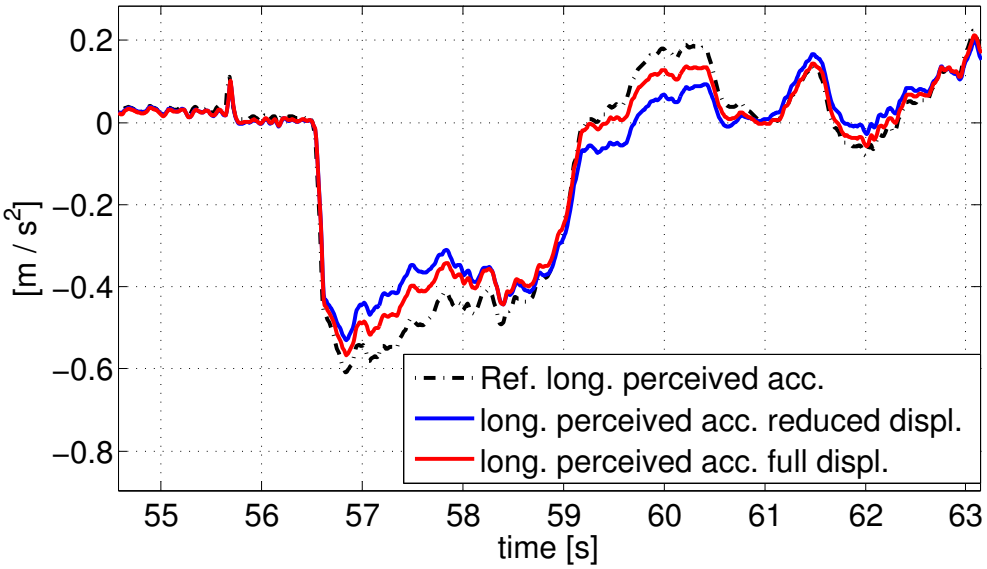


Figure 4.10: Perceived acceleration tracking, reduced longitudinal displacement (tuning effect).

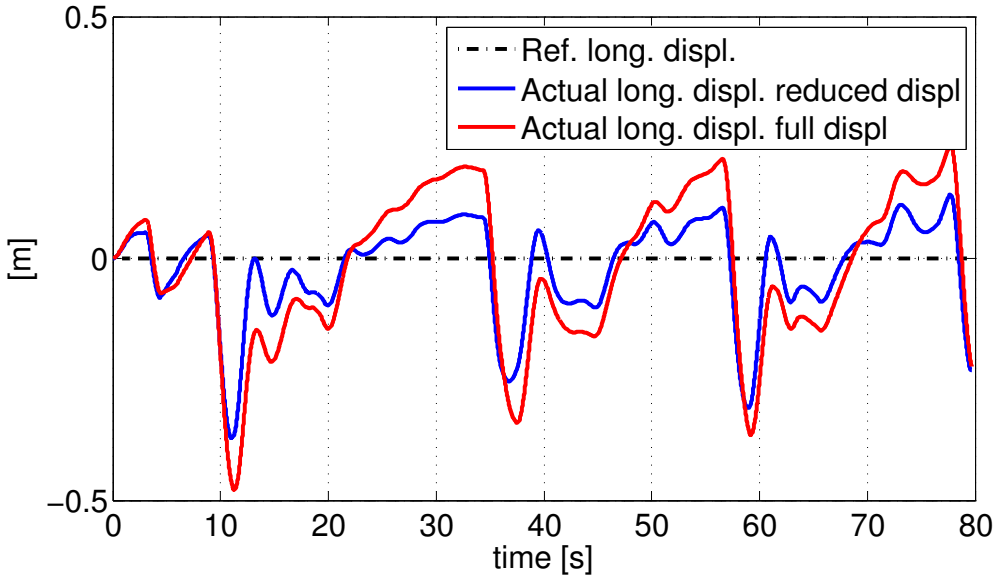


Figure 4.11: Reduced longitudinal displacement (tuning effect).

slot, it is clear that an error is introduced, hence this parameter has to be carefully set, depending on the particular track involved and the ability of the driver (a non professional driver will have reduced dynamics and a lower prediction time could be more suitable, and viceversa). The effect of a constant reference signals for prediction can be thought as having a sort of "conservative" effect on the computed controls. With large N_p, at each step the algorithm is instructed to find a compromise to achieve tracking within

an interval long enough to allow a smarter exploitation of the platform. In this sense, tilt coordination is better exploited and linear displacements are limited, so that more working area will be available for possible future needs. If N_p is too small, the signals over the prediction time are very close to the real future value, hence a better tracking is provided, but this comes at the cost of an increased probability of saturating the actuators, especially the linear ones (having a more prompt time response, lower look-ahead time will make the algorithm prefer this kind of actuation): the consequence is the infeasibility of the problem. In figure 4.13 the effects of an increase and a reduction on that value are shown. We can see that increasing N_p (+50%), the tracking performance is not significantly affected in terms of perceived acceleration but the platform exploitation is quite different, with a more intensive use of pitch rotations. On the other hand, a small prediction window (−50%) improves the tracking quality but at the cost of safety: at 11 s the problem becomes infeasible and the procedure is stopped. Note how in this latter case tilt coordination is disregarded by the algorithm.

In any case, it is worth emphasize the fact that the value of N_p directly affects the resolution time, hence it has to be chosen as small as possible to this regard, while assuring the convergence of the optimization.

Remark 4.3.1. Note that within the implemented setup, each subsystem requires approximately 500 μs to be solved, hence the real-time constraint is fulfilled.

4.3.3 Other motion subsystems

The analysis of the longitudinal-pitch subsystem is significant to understand the effectiveness of the proposed MPC method; in any case, for the sake of completeness we present here an overview about the performance of the other subsystems.

Lateral acceleration and roll rotation are coupled as well as longitudinal and pitch: in figure 4.14 and 4.15 samples of these signals are shown. Note that for quality of the simulation, lateral acceleration is important for reproducing the sensation of the “nervous” and abrupt fast dynamics that characterize the category of cars considered during turns insertions: the acceleration has quick variation from high positive values to negative ones. To the contrary, in this case the tilt coordination contribution has less importance: low frequency accelerations in the lateral direction are less frequent, and mainly due to particular track features (e.g. parabolic curves). Moreover, introducing a remarkable tilt coordination in this direction could can be dangerous, e.g. in the presence of fast chicanes where the necessity of fast repositioning makes the platform behaviour incompatible with the low-velocity perceptive thresholds and is likely to introduce considerable false cues with respect to the visual system (motion sickness). The effect of this consideration can

be seen in figures 4.16 (lateral displacement) and 4.17 (roll displacement), where the angular position is kept at low values. Note the higher frequency in this latter signal, due to the reproduction of the impact of the track on the car suspensions related to track imperfections, kerbs crossing and so on. Note in particular that when ϕ reaches high values, they are likely to be "held" for a little time (with small, high frequency signals superimposed): the effect is due to the simulation of a kerb crossing, a manoeuver that force the vehicle to remain tilted until completed. In this subsystem the rotational DOF is then more important for reproducing real features of the motion rather than "apparent" ones.

Performance in yaw and vertical DOFs are now analyzed. As seen, these subsystems are completely decoupled and don't affect, neither are affected by the behaviour along the other DOFs. Figures 4.18 and 4.19 represent the perceived yaw velocity and the platform displacement for a sample of the telemetry. Yaw is particularly significant in the considered testbed having a great operational area. The results show the great exploitation of the platform capabilities, useful to reproduce the perceptions along the track turns. Observing the simulator in action, the yaw behaviour showed another interesting feature: in certain points of the circuit, after fast turns it happened that the algorithm forced the platform to stay in an "intermediate state" between the maximum displacement reached and the zero-position. This was a clear example of the prediction effect: the algorithm naturally handles the platform position to allow for better exploitations, in case there could be another turn. After a while, the platform was forced to regain the central position with slow and smooth movement along yaw direction, a clear effect of the Washout Action. Hence, MPC permits a better Washout in this sense too, "smartly" delaying its effects. The z direction shows the results in figures 4.20 (sensed acceleration) and 4.21 (physical displacement). This degree of freedom has both low and high frequency components: the former help the driver in perceiving the varying altitude of the track (useful to train drivers to new tracks), the latter, combined with the aforementioned pitch effect, contribute to reproduce the sensations tied to the track conditions and the consequent effects of the chosen components of the car (this feature can be useful to perform virtual test on new-designed components, as suspensions or break, as well as different set-ups, avoiding expensive on-road tests). Note that to exploit at best this capabilities, the track has to be carefully reproduced (e.g. with up-to-date laser detections).

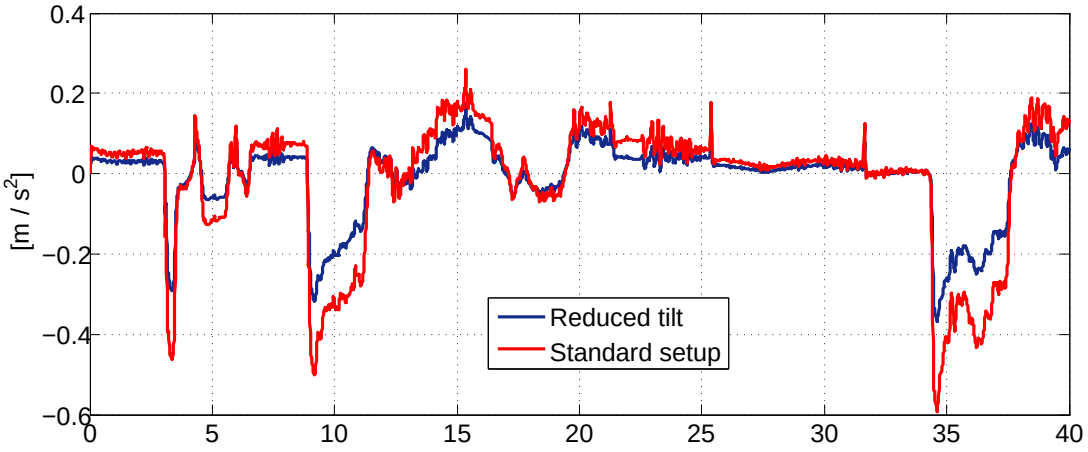
4.3.4 Using the prediction: simulation results

We want now to test the possible improvement if a valid reference signal is provided for the prediction, i.e. not constant references. Notice that this is a preliminary study done

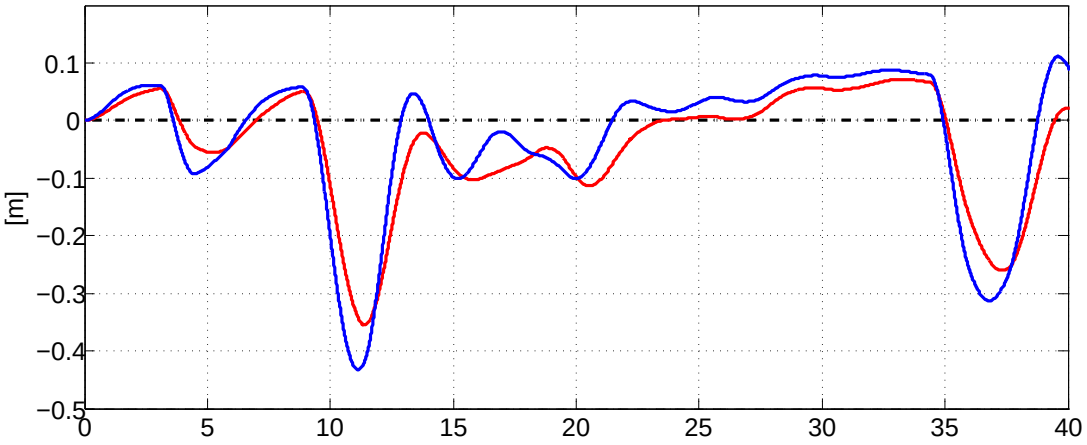
by resorting to simulations only, i.e., the algorithm has not been tested on the platform yet. It is assumed that a telemetry reference signal is available, hypothesis that can not hold true in the a context. In figure 4.22 a prediction over a time interval of 8 s is used. The tracking of perceived acceleration gives very good results, with optimal performance and no sign of wrong cues, e.g. motion inversion is absent. Of particular relevance is the exploitation of the working area: the knowledge of a reliable reference together with a "wide enough" prediction horizon allows the algorithm for a maximization of the platform capabilities, and this is revealed by the reduced displacement both in longitudinal and pitch DOFs. In fact, prediction lets the procedure know in advance the behaviour of the signal, hence the movements can be reproduced in an optimal way and does not need for large displacements: in this sense, by lowering the weights in a smart way it would be possible to improve the dynamics of the device.

Beside the need of the reference to track, using a non-constant prediction has the drawback of increasing drastically the computing time for the optimal problem: as an example, the previous shown results took about 410 seconds to run on a 60 seconds telemetry on an I7, 2.67 GHz PC using MATLAB and `qpOASES` for the longitudinal-pitch subsystem only. Following the idea proposed by Longo, Kerrigan, Ling, and Constantinides (2011), a *blocking* strategy is introduced. The mechanism is simple: a wide prediction horizon requires a large number of samples, hence the increase of N_p makes the problem very hard to be handled in real time, because of its size. For the reported example, 8 seconds at 100 Hz means $N_p = 800$, with respect to $N_p = 30$ used for the constant prediction case. To reduce the burden, the blocking philosophy proposes to divide the prediction horizon in different segments, with increasing sampling time. In this way, by considering for instance 2 s at 100 Hz and 6 s at 20 Hz, a 8 seconds prediction windows corresponds to $N_p = 230$, $3 \div 4$ time smaller than with the usual predictive strategy. Figure 4.23 depicts the results obtained by implementing this approach with the reported blocking setup, compared to the classical predictive approach. The tracking performance is almost identical, and the pitch displacement as well; only the longitudinal displacement slightly differs, but doesn't seem to have a negative impact on the working area exploitation. In this case, the calculation time was about 35 seconds for the same 60 s telemetry, with a medium optimal resolution time of 5 ms, on the same machine: beside being a single subsystem, this strategy already works in real-time. Note that the weights have to be adapted when blocking takes place, since lower control frequency means a proportional increase in the weights to preserve the consistency of the problem.

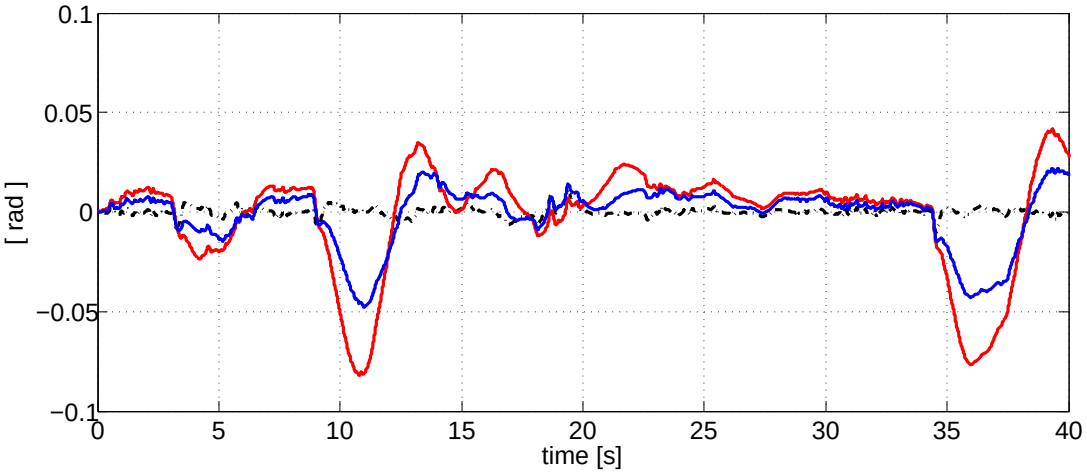
These results are promising, but a strategy to compute real-time computation of reliable references has to be studied.



(a) Perceived acceleration tracking.

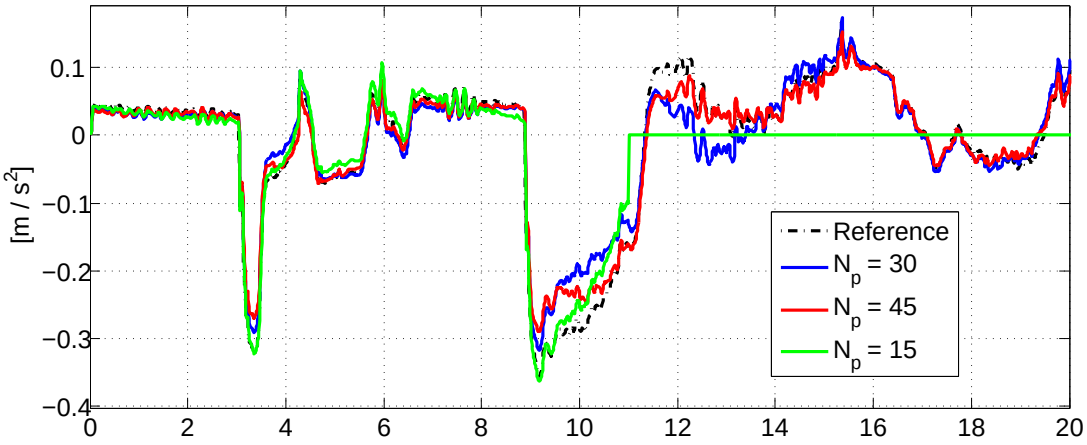


(b) Longitudinal displacement.

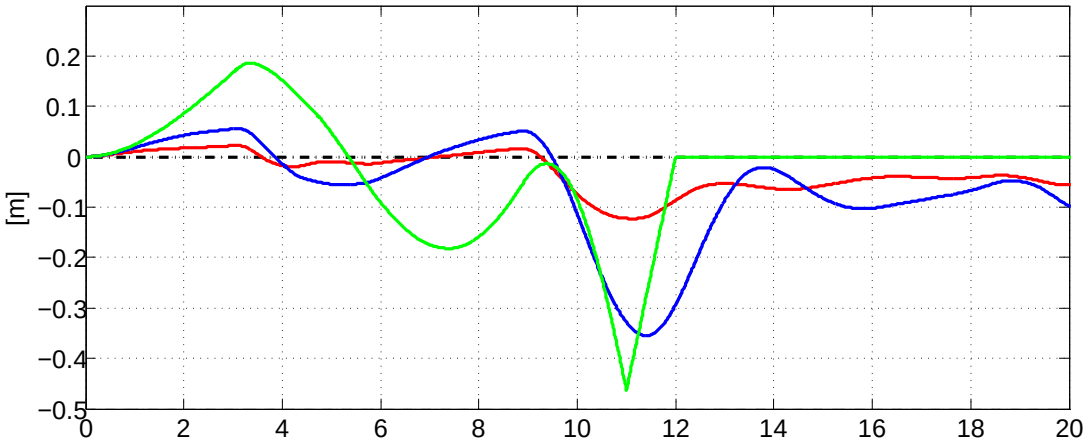


(c) Pitch displacement.

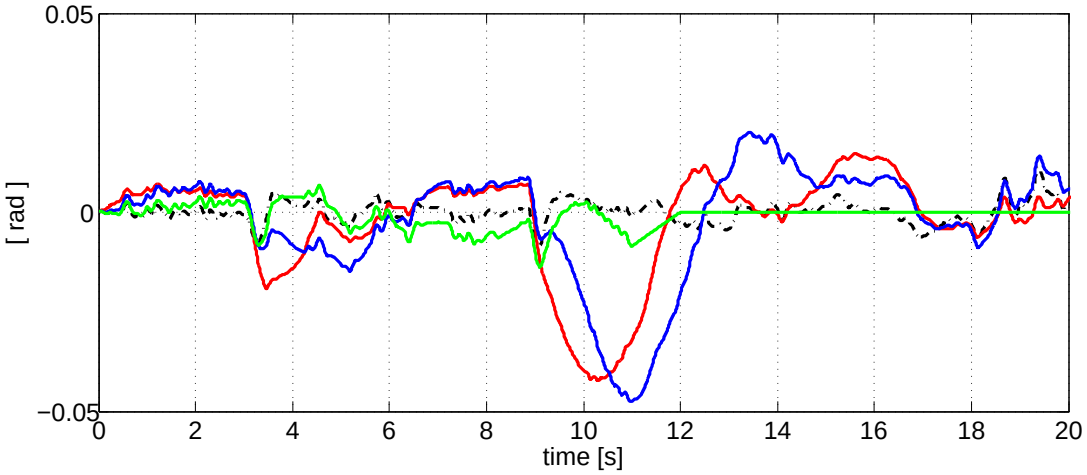
Figure 4.12: Comparison with different setup of tilt coordination: the tilt coordination is reduced, while longitudinal displacement is unaffected. Weights have to be adapted to preserve feasibility.



(a) Perceived acceleration tracking.



(b) Longitudinal displacement.



(c) Pitch displacement.

Figure 4.13: Comparison with different N_p values: an increase limits platform exploitation (more conservative control signal), a decrease makes the system faster but less robust.

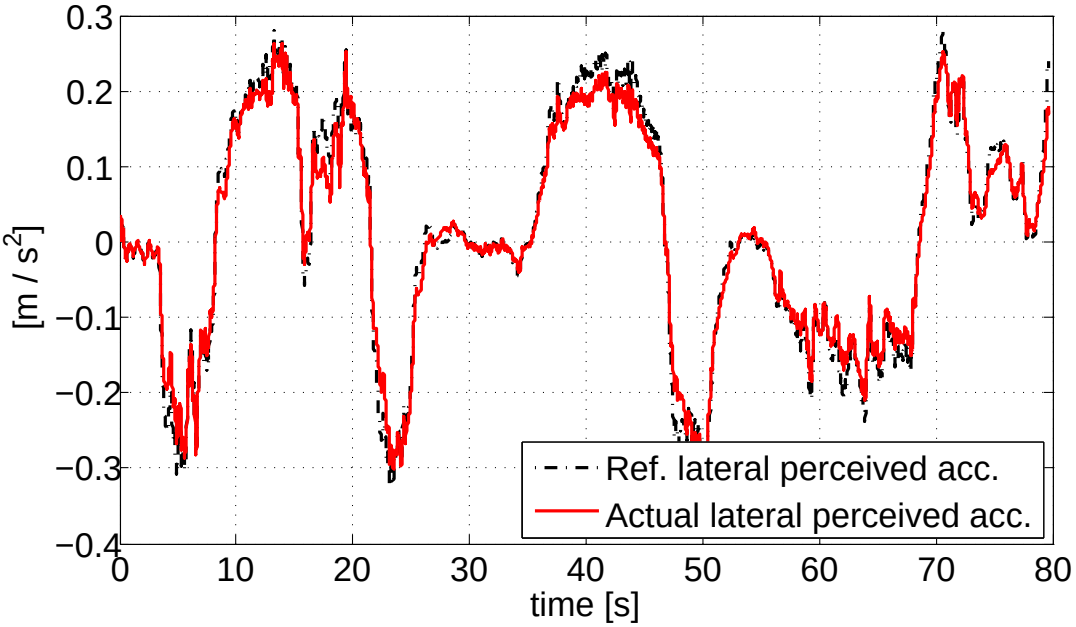


Figure 4.14: Perceived lateral acceleration tracking with the best tuning configuration.

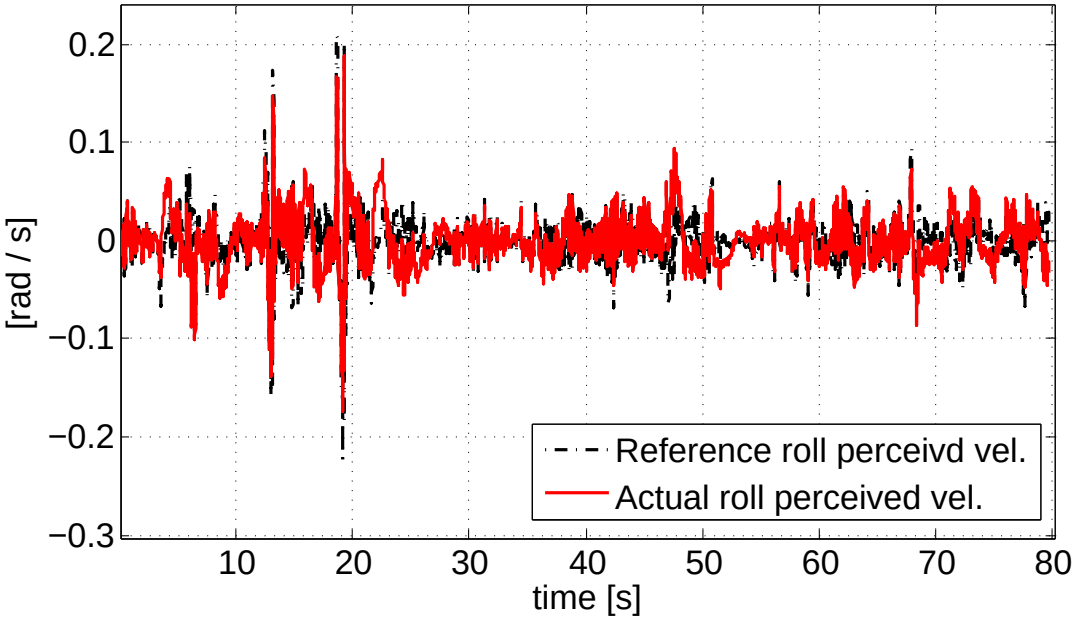


Figure 4.15: Perceived roll velocity tracking with the best tuning configuration.

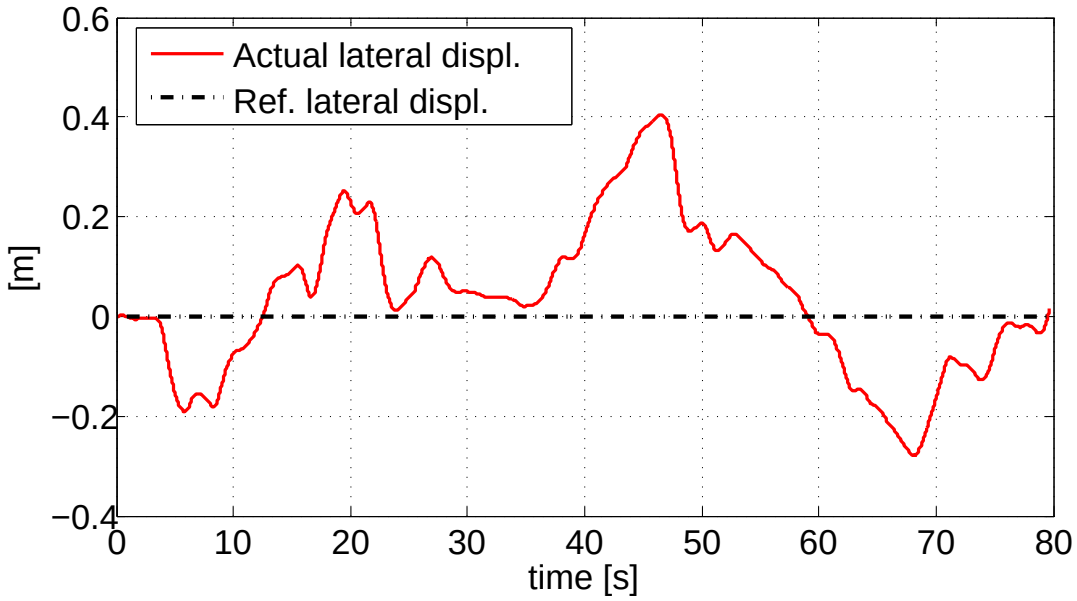


Figure 4.16: Lateral position displacement with the best tuning configuration.

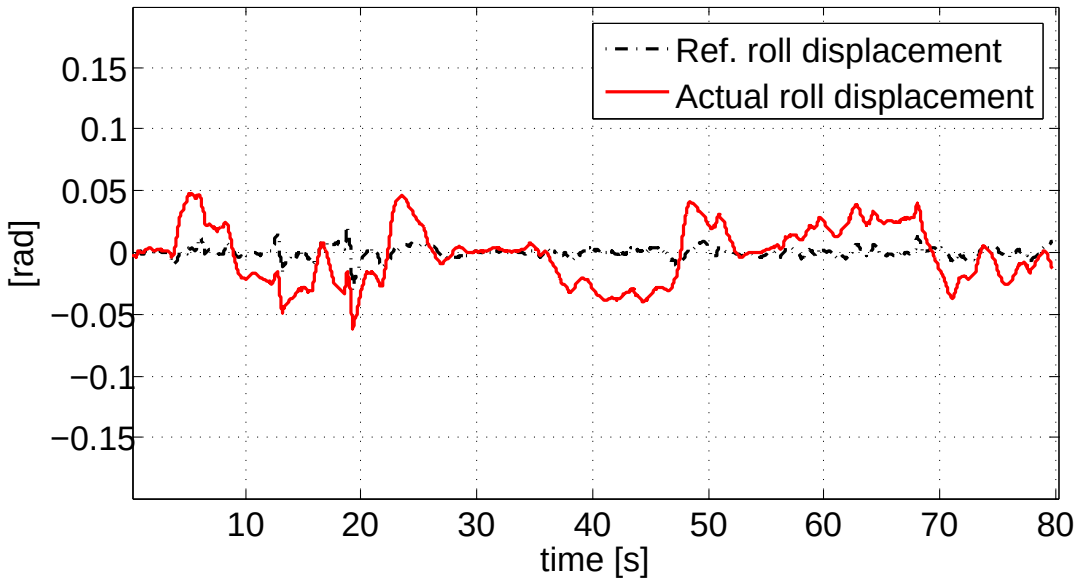


Figure 4.17: Roll displacement with the best tuning configuration.

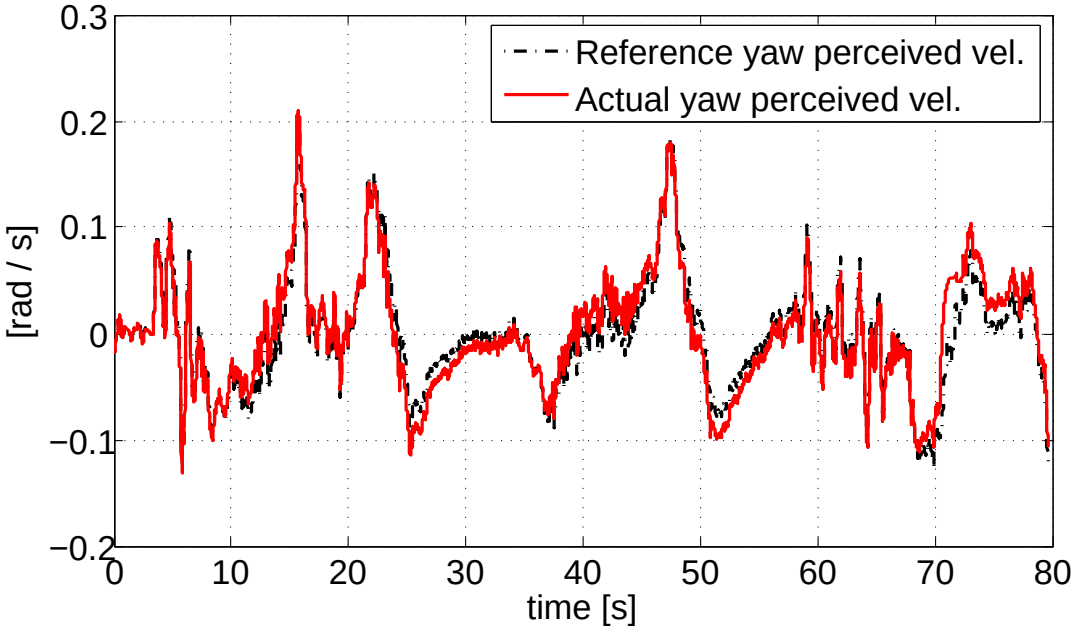


Figure 4.18: Perceived yaw velocity tracking with the best tuning configuration.

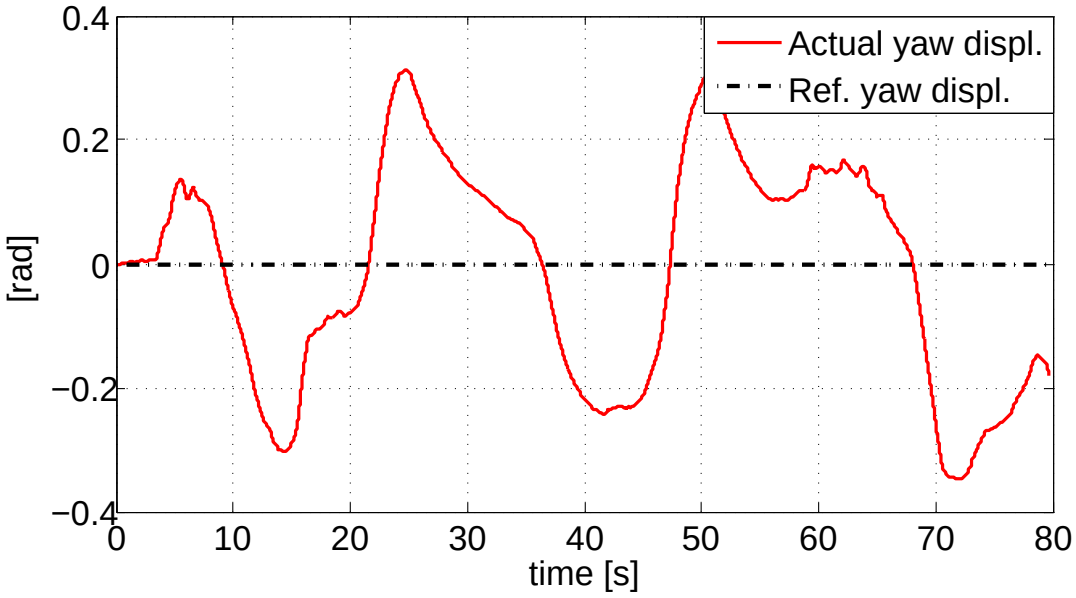


Figure 4.19: Yaw displacement with the best tuning configuration.

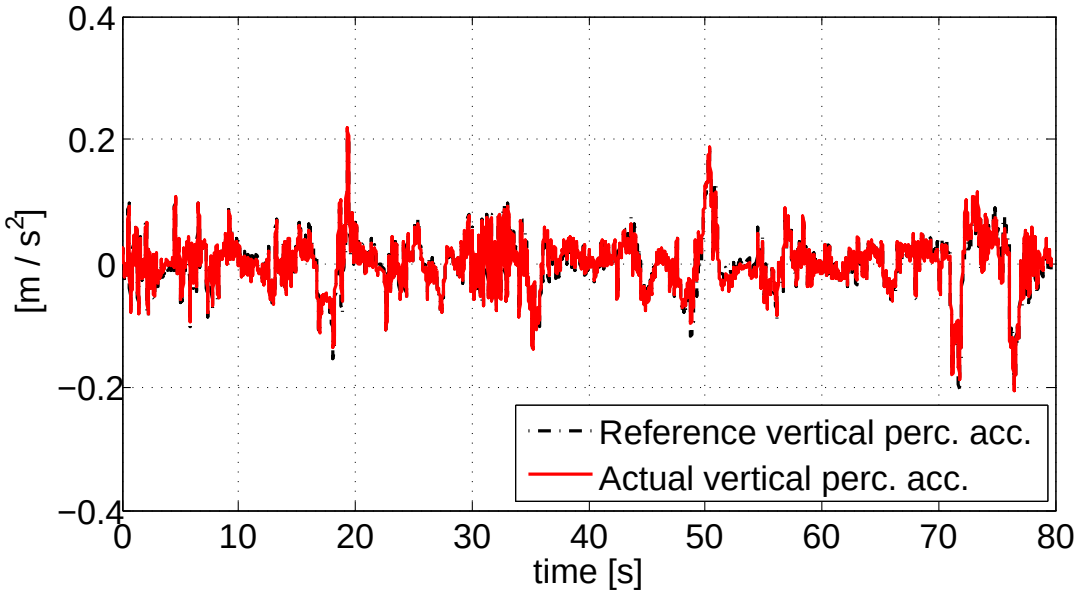


Figure 4.20: Perceived vertical acceleration tracking with the best tuning configuration.

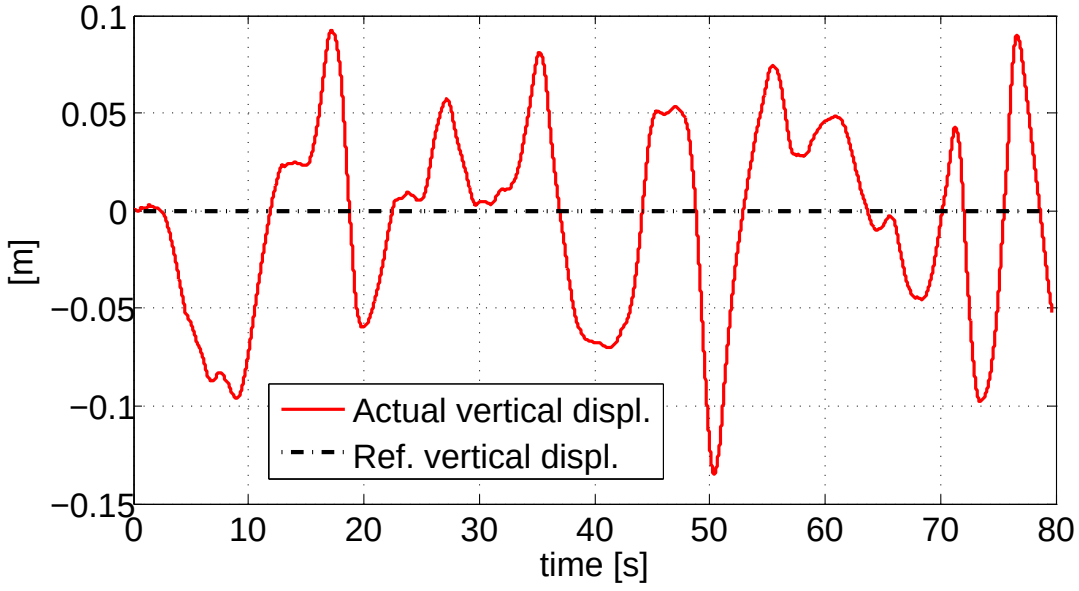
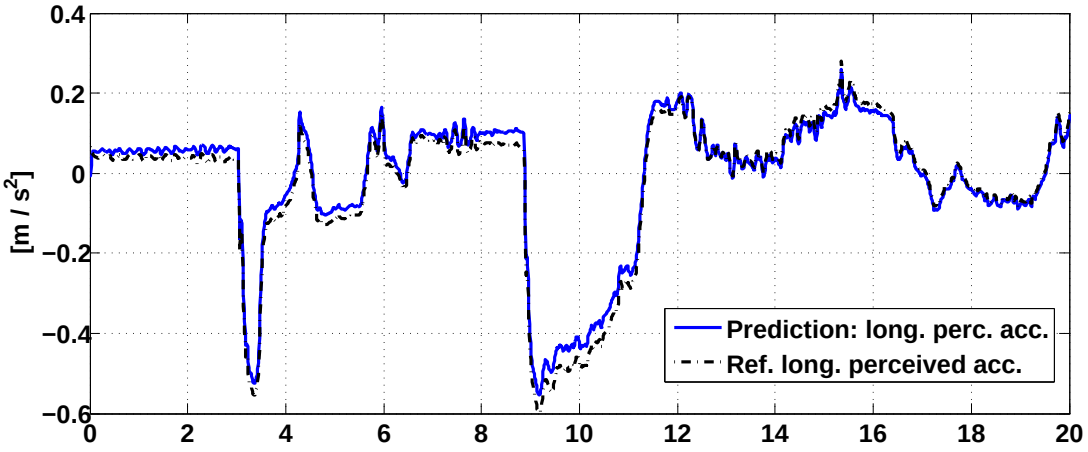
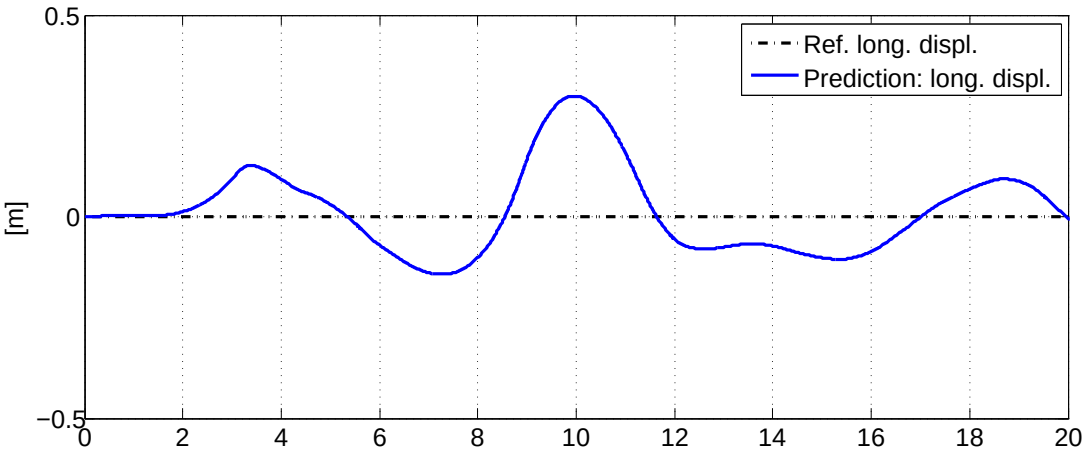


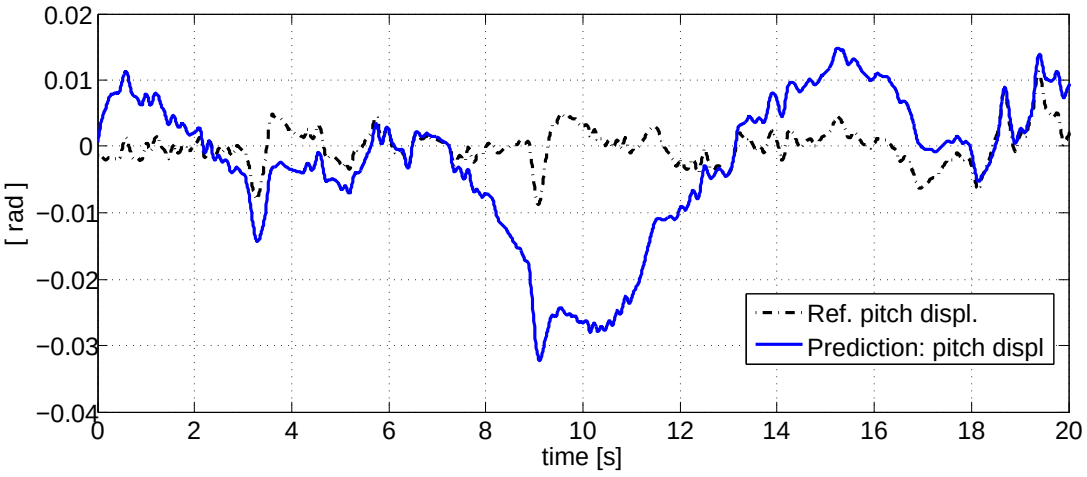
Figure 4.21: Vertical position displacement with the best tuning configuration.



(a) Perceived acceleration tracking.

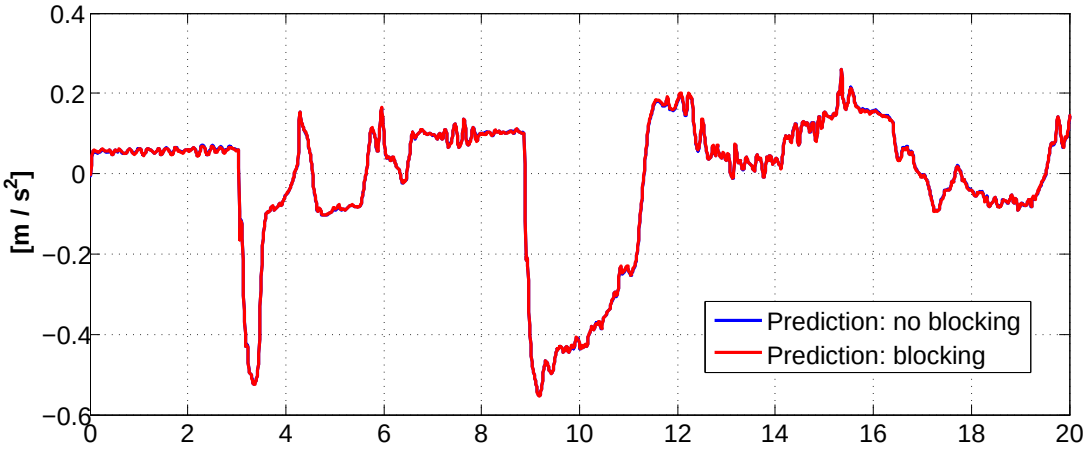


(b) Longitudinal displacement.

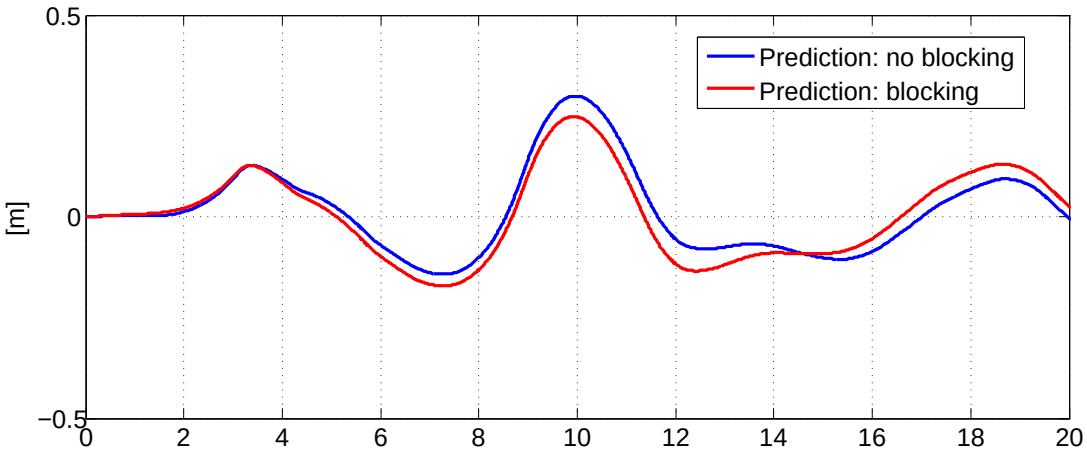


(c) Pitch displacement.

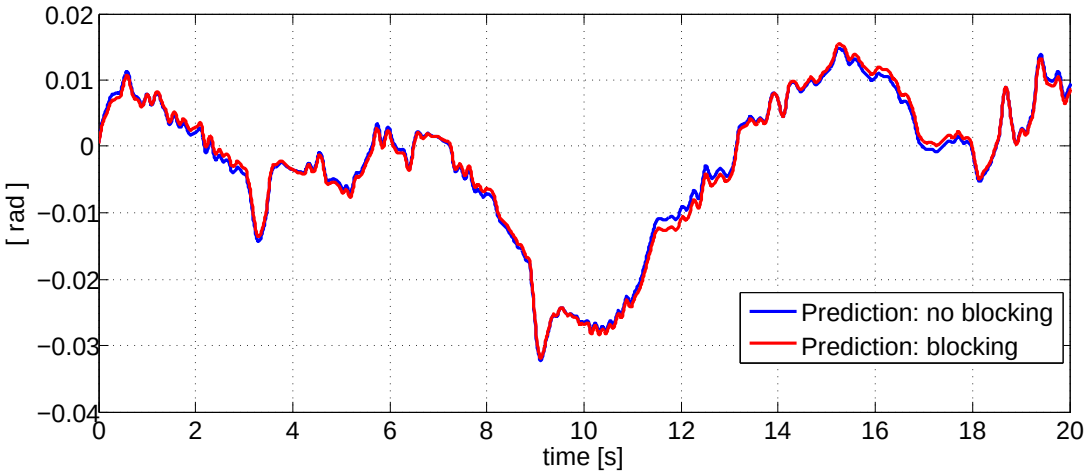
Figure 4.22: Prediction results: despite the larger computation time, the tracking performance is optimal with minimal use of the working area, that could afford a less conservative setup.



(a) Perceived acceleration tracking.



(b) Longitudinal displacement.



(c) Pitch displacement.

Figure 4.23: Comparison of prediction performance with and without *blocking*: the results are comparable and suggests this approach to be valuable if a reliable reference is available.

4.4 MPC hardware implementation on FPGA

Up to now, the importance of real-time performance for the Motion Cueing algorithm has been stressed several times, and this is the most critical point with MPC. The proposed approach has been able to reach the desired behaviour with some intuitions, like model decoupling, ad-hoc setup of the solver and so on. In any case, other devices may require different approaches: as an example, classic hexapodal structures cannot take advantage of model separation, and more accurate modelizations of the mechanical system could lead to much more complicated models and constraints. On the other hand, the performance of VI-DRIVESIM could be improved by increasing the control rate and/or the model complexity, as well as introducing the prediction. These observations lead to consider a *hardware implementation* of the MPC paradigm.

The ideal hardware technology for this stage should be *reconfigurable*, in order to allow for easy modification and update. In this sense the *Field-Programmable Gate Array* (FPGA, figure 4.24) stands as a perfectly suitable platform for scientific computation with the aim of deploying MPC for fast dynamic systems. FPGAs are a good alternative to

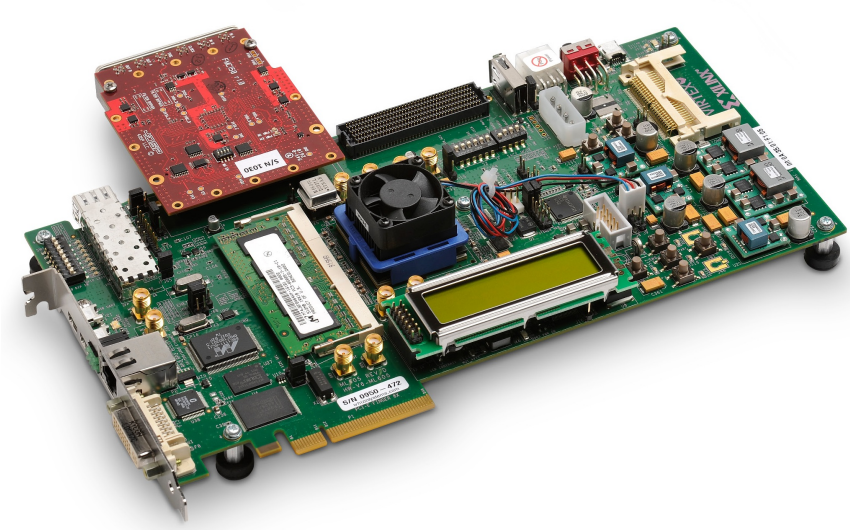


Figure 4.24: Example of FPGA: XILINX Virtex 6.

Application Specific Integrated Circuits (ASICs) for embedded MPC applications having a cheaper cost for small volumes hardware, greater flexibility, and a shorter design cycle, reducing the risk while still maintaining a high-power efficiency. First works on FPGA suitability for constrained MPC were by Ling, Yue, and Maciejowski (2006), and different practical applications have been proposed over the years (e.g. by Hartley, Jerez, Suardi,

Maciejowski, Kerrigan, and Constantinides (2012)). One of the essential points is the choice of the optimizing method. Recalling the classification in section 3.4, the choice should be between Active Set and Interior Point methods. Lau, Yue, Ling, and Maciejowski (2009) give an insight on the topic, as well as Jerez, Ling, Constantinides, and Kerrigan (2012): while Active Set can be appropriate for low-dimensional problems, in the situation at hand the best choice is Interior Point. First, AS has a worst-case complexity increasing exponentially with the problem size, and in a context (floating point, single precision) where the complexity in terms of calculations and cost is proportional to the number of arithmetic operations (multiplication and division), having a polynomial complexity bound is preferable. Moreover, with AS the size of the linear systems that need to be solved at each iteration changes depending on which constraints are active at any given time and in a hardware implementation this is problematic, since all iterations need to be executed on the same fixed architecture. IP maintains a constant predictable structure and is more suitable in this field. It is important to emphasize the fact that if the solving algorithm is not properly chosen (and implemented), leading to a greater number of arithmetic operations, the only way to improve the performance is to buy more powerful hardware devices, which are much more expensive (the cost grows exponentially with the size , with an estimated factor between 2 and 4).

By exploiting the structure of the problem, the procedure can be parallelized, determining acceleration of QP solvers for linear MPC. First investigations has been made by Ling, Yue, and Maciejowski (2008), and the pipelining capabilities of FPGAs has been studied in later works (Jerez, Ling, Constantinides, and Kerrigan, 2011a; Jerez et al., 2012).

Given the characteristics of the present work about MPC for Motion Cueing, and in particular the features of the testbed, this field of research has been seen with deep interest: an hardware implementation would be an improvement per se, and the possibilities of parallelizing the problem fit perfectly the practical needs. This part of the research has seen the collaboration of the Control and Power Group of the Department of Electric and Electronic Engineering from Imperial College London, in particular Dr. Eric Kerrigan[1] an Juan L. Jerez[2]. The idea of the converting our implementation to FPGA follows what shown in Jerez, Constantinides, and Kerrigan (2011b). Based on primal/dual IP method (Boyd and Vandenberghe, 2004), the insight is the reorganization of the matrices coefficients to achieve a particular sparse structure, the *banded* structure: this allows for minimum residual method (Bernd, 2011) to be applied for the resolution of the consequent linear systems, which has a solid, parallelized implementation in FPGAs. Another block

[1] e.kerrigan@imperial.ac.uk
[2] juan.jerez-fullana@imperial.ac.uk

on the same device will handle at the same time the problem reformulation to enhance performance and exploit all the capabilities of the FPGA: note that this hardware gives the best performance when all its components are used at the same moment (in terms of computational/power consumption ratio). Memorization is an issue on the FPGA, since upgrades can be expensive. Dealing carefully with the input/output latency and throughput (which can be calculated a priori in a full-operational FPGA), this aspect can be improved by interfacing the device with an external source where the data can be picked at each iteration. In this way the cost can be limited without losing too much speed. Note that in any case the sparsity of the matrices allows for efficient data compression. More, with different problems of sufficiently small size, the components for solving each of them can be implemented on the same board and run independently at the same time.

At the time of writing, there doesn't exist a proper FPGA implementation of the problem, but studies about the supposed computing time have been made. If we consider a board with 3000 "units" of resource, the timing would be ca. 250 μs per iteration of one QP problem for the complete model (33 states) and ca. 30-40 μs per iteration for each of the four subproblems. Such FPGA would be quite expensive (as an example, the XC6VLX240T FPGA, for which there exist 1500$ development boards, would have around 750 of "resource", and the cost increase -as well as decrease- nonlinearly with this quantity). As a rough approximation, the computation time is expected to double by halving the resources and viceversa. The research is in progress and first steps for the hardware implementation have been executed.

5
Conclusions

In this first part of the dissertation, the design of an innovative Motion Cueing algorithm based on MPC techniques has been presented. The use of the MPC has three fundamental motivations:

1. to exploit a model of the human vestibular system to compare the in-vehicle and the in-platform perception;
2. to adopt a suitable technique to efficiently handle hard constraints;
3. to take advantage of a time-domain control that ease the comprehension of the problem and the consequent regulations.

The proposed algorithm represents a novel approach to motion cueing that completely changes the classic paradigms of washout filters: tilt coordination and working area constraints are handled through an optimization procedure without the employment of any filter.

The integration of the perceptive models strongly improves the Motion Cueing effects, helping to prevent incoherent behaviours and allowing to reproduce effective signals to track for the motion controller. The presence of constraints improves the exploitation of the working area without losing physical reliability, enhancing the overall robustness of the problem.

A distinctive point is the ease of tuning procedure. This is a fundamental improvement with respect to the standards in this field, where the lack of "physical connection" between the parameters and the motion behaviour, as well as the complicated integration of the passive filters and the lack of capabilities to deal a-priori with the limitations on the working area made the MC hard to tune and required specialized (and expensive) experts to handle the problems, making the adoption of dynamic simulators difficult in various field. In particular, this approach makes possible the direct interaction with the drivers, in the sense that their feedbacks during the tests can be immediately translated into parameters modification, with a clear improvement in the regulation time. This feature is crucial, as it is been verified that one of the major problems in driving simulators is the *adaptation*: after a certain time on the device, the driver adapts it style and perception to the virtual environment even if it does not reflect the reality (Straus, 2005). This is a problem indeed, hence a fast regulation of the system is necessary.

It is worth noting that, given the high system dimension, although implementing a real time MPC procedure is not a trivial task, the algorithm works at 100 Hz control frequency. This real time capabilities, improved by the explained approximation, made possible the implementation on a small size dynamic driving simulator, namely VI-DRIVESIM, validating the tuning methodology and the performance improvement in a practical environment. Feedbacks from professional test drivers have been encouraging.

The natural development will be the introduction of prediction. As seen, the difficulties in terms of computational time can be overcome by using a *blocking* strategy, but the need for reliable references is an issue. One possible idea is to exploit the availability of a virtual driver (Frezza, Saccon, and Bacchet, 2003; Frezza and Beghi, 2006) to obtain a reliable future trajectory. This information can be integrated in a *stochastic* variant of MPC, and one possible approach could the *tube-based MPC* (Langson, Chryssochoos, Raković, and Mayne, 2004), where uncertainties are taken into account in the problem resolution. A possible improvement can be the introduction of a *learning* procedure, where the algorithm starts by relying its predictive information on the virtual driver, and then adapts itself to the specific user and to its driving style. The first insight about prediction, whose simulation results have been reported, showed encouraging results.

The need for fast optimization would be critical in presence of prediction, and the development of FPGA solutions comes at hand. The first simulation tests are being carried on at the time of writing, and possible improvement in term of speed and cost are being examined. In particular, the newly approach on *first order methods* seems promising (Jerez, Goulart, Richter, Constantinides, Kerrigan, and Morari, 2013): if the constraints on the original problem can be reformulated as to depend only on states $\mathbf{x}$, inputs $\mathbf{u}$ and their (approximate) derivative $\Delta\mathbf{u}$, it is estimated that this approach would require about 1350 resource units, and the time per iteration would be $3-4$ μs for the complete model and $1-2$ μs for the subproblems, a significant improvement.

Part II

Control Techniques for a Hybrid Sport Motorcycle

1 Introduction

Transportation systems and efficient energy utilization are two of the most relevant research topics on a world-wide scale, for their economical and environmental impact. In the European context European Union (2011), *fossil fuels*, despite their high levels of pollution, are still extensively used for energy production and transportation. To assure sustainability and more confidence on energy supply, the European Union has studied a careful energetic policy for the next ten years, whose aim is to reduce *greenhouse gas levels* and *energy consumption* by 20% and increase the share of renewables by 20%, all these three targets by 2020.

However, the analysis of current trends show that the second target will not be reached. One of the main causes of such phenomenon is associated with the intense use of energy for *transportation*. Consequently, there is an increasing level of attention on the development of hybrid vehicles, that can help in addressing both the reduction of greenhouse gas levels and the increase of energy efficiency of transportation systems,

hence capturing the interest of international research and vehicles manufacturers.

In general, a hybrid vehicle couples two different propulsion systems, which are made to coexist mainly to *improve performance* and *reduce emissions*. Although various typologies have been studied, the most widespread is the *electric hybrid*, which couples an internal combustion engine to an electric machine powered by the energy stored in dedicated systems (*accumulators*). Depending on the implemented architecture, hybrid electric vehicles are divided into three main categories (Chan, 2007; Chan, Bouscayrol, and Chen, 2010; Ehsani, Gao, and Miller, 2007; Emadi, Rajashekara, Williamson, and Lukic, 2005)

Series Hybrid *Hybrid electric vehicles* (HEV) of type *series*, or *Range Extended Vehicle* (REV), exploit a combustion engine to generate the energy needed to recharge the batteries and to power the electric machine that provides all the motion power (schematics in figure 1.1). The Internal Combustion Engine (ICE), which is not responsible for traction, is forced to operate around the point of maximum efficiency in all driving conditions, by starting it in the case where the state of charge of the battery reaches the established minimum value, and turning it off to the achievement of the maximum charge level. As made clear by the architectural scheme, there is

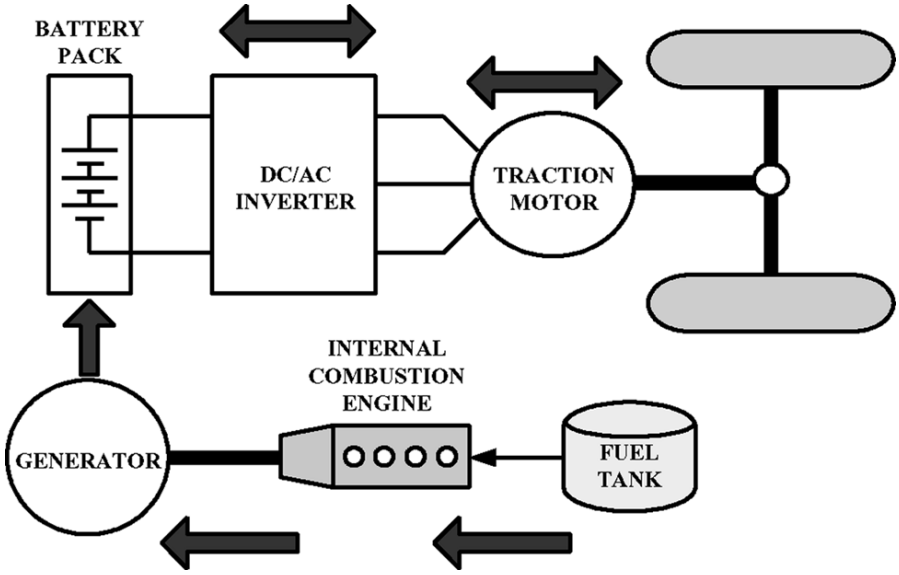


Figure 1.1: Series Hybrid schematics.

not a mechanical connection between the ICE and the wheels: in this way it can operate in optimum conditions of speed and torque to provide only charging power, limiting consumption of fuel and increasing the efficiency. The weak point of this type of hybrid vehicles is the presence of two *dissipative* energy conversion, from

combustion engine to generator and from generator to electric motor. A further disadvantage is the need to size both electrical machines (generator and motor) with respect to the maximum power of the vehicle, which is useful only in rare circumstances, hence making the dimensioning a challenging task. This architecture can be useful in all those contexts in which the driving conditions are clearly defined, such as the urban environment, or if applied to vehicles whose dimensions allow the positioning of the electrical machines which, by acting as a sole source of traction, are very massive (locomotives, buses, trucks, etc.).

Parallel Hybrid HEV in *parallel* configuration are equipped with an internal combustion engine and an electric machine, *both* connected to the drive shaft, which can operate simultaneously or individually depending on driving conditions (figure 1.2). Usually while driving at low speeds, such as in urban environments, the more efficient electric motor is used, while at high speeds ICE is preferred, using the electric machine as a generator to recharge the batteries (with the help of regenerative braking) or to provide boost contributions. To improve efficiency, the parallel HEV are often

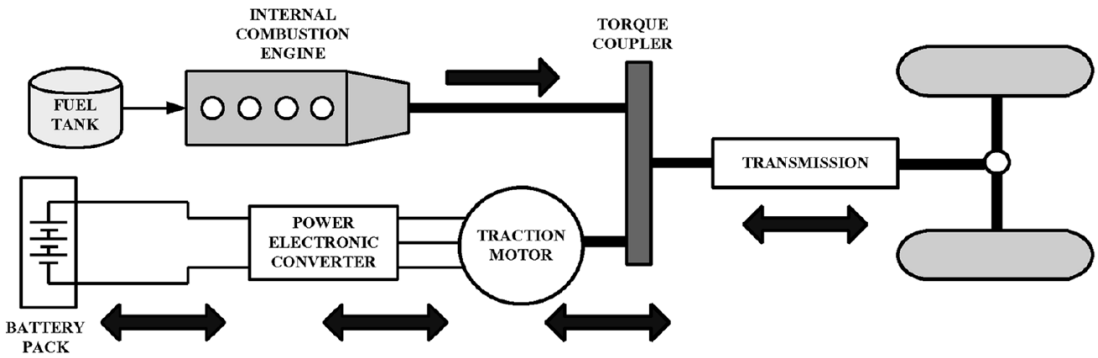


Figure 1.2: Parallel Hybrid schematics.

equipped with *continuously variable transmissions* (CVTs), that allow to optimize the operating point of the internal combustion engine for every possible torque request, hence improving the efficiency and the fuel consumption. The main advantages of the parallel configuration are

1. the presence of a *single energy conversion*, which allows to reduce the dissipation;
2. the use of a single electric machine of *small size*, which allows to reduce costs and space needed for its positioning.

Since it is not the main source of traction, the electric motor is dimensioned to

provide a lower power than the maximum for the vehicle, in this sense this class can be further divided into three categories:

1. *micro hybrid*: equipped with an electric machine of limited voltage and power ($V < 60$ V, $P < 5$ kW), it replaces the alternator, while the propulsion is guaranteed by the internal combustion engine. The electric motor is used for the *start and stop* system, which shuts down the engine when it is not used (as for example at traffic lights) allowing a saving of fuel between 2% to 10%, in the urban cycle. The electrical machine allows a smoother start compared to normal starters and a limited regenerative braking, since it is connected to the motor shaft by means of *belts* (*Belted Starter Alternator* – BAS).
2. *mild hybrid*: these are an evolution of micro hybrid vehicles, using medium voltage and power devices ($V > 100 \div 200$ V and $P > 10 \div 20$ kW). The goal is to get a boost effect using the electric machine to provide an additional torque to the endothermic propulsion during acceleration or braking. This type of vehicle allows to enhance the regenerative braking and, if possible, the operation of the vehicle in "pure-electric" mode at low speeds and loads, with a fuel saving between 10% to 20% in the urban cycle.
3. *full hybrid*: they are equipped with high voltage and power devices ($V > 300$ V and $P > 50$ kW), which can provide the power required to propel the vehicle. In this way it can travel in purely electric mode (*zero emission vehicle*), limited only by the capacity of the batteries, as well as with the internal combustion engine only or a combination of the two. This particular type of vehicles ensures a fuel saving between 20% and 50% in the urban cycle.

Series – Parallel Hybrid HEV of type *series-parallel* (figure 1.1) are a combination of the two previous architectures obtained using a *planetary* transmission that allows to decouple the vehicle speed ω_V from the electric machine and the internal combustion engine ones, respectively ω_E and ω_I. The electric motor can then be used to perform simultaneously the propulsion of the vehicle and the charging of the batteries (*power split*) if $\omega_E < 0$, or the optimization of the working point of the internal combustion engine if $\omega_E > 0$, thus decreasing fuel consumption. In addition, by exploiting a particular locking system, it is possible to disable the electrical machine to obtain a parallel configuration or, by absorbing all the torque provided by the ICE, to operate in series configuration. It is easy to understand that the growing architectural complexity requires sophisticated control systems, making this type of vehicles considerably more expensive than previous ones.

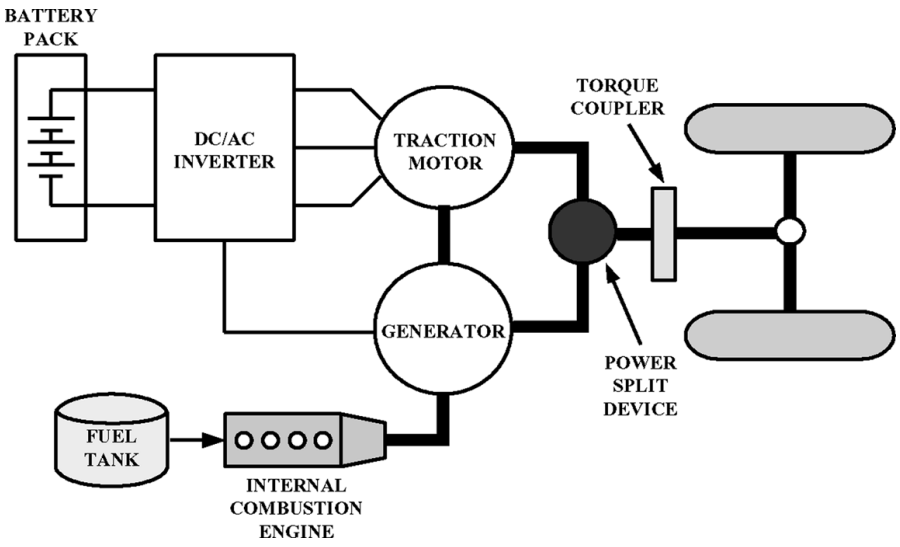


Figure 1.3: Series-Parallel Hybrid schematics.

While many models of hybrid cars have been developed and put on the market in the last few years, hybrid scooters and motorcycles have not seen a large scale production yet, despite their promising peculiarities in terms of fuel economy and environmental impact, and mobility capability. One of the main obstacles to their spreading is the limited autonomy of batteries, an even greater problem for two-wheeled vehicles given the limited available space. The presence of the electric engine and the battery pack makes the problem particularly challenging with motorcycles, since it may affect the rider's driving feeling with respect to an equivalent model provided with an internal combustion engine only. In this sense, an appropriate design of the *Power Management System* is crucial to deal with the requirements of acceptable autonomy of the electric engine and preservation of satisfactory driving feelings, which depends most on the strategy that regulates the instantaneous coupling of the two propulsions.

To this respect, it is important to have a means of evaluating the dynamic behaviour of the motorcycle when it undergoes the hybridization step. It's easy to understand that the presence of an electric machine on the motorcycle can highly modify the vehicle behaviour, e.g. due to the increase in weight and the effects of the additional torque provided by the electric engine. In this context, a *virtual environment* capable of accurately describing the vehicle dynamics is a key feature for testing different engine and power systems management algorithms and their impact on the vehicle manoeuvrability. Moreover, one of the crucial design steps, namely the sizing of the batteries, is usually carried out with static simulations. A standard telemetry is processed off-line, boost and charge steps are identified and *point-wise* energy consumption/recovery values are calculated. Reliability

for this kind of analysis is not assured, since the effects of the boost phase on the vehicle stability and performance is not investigated. A *dynamical simulation* environment can therefore be effective also in this design stage.

In this part of the thesis, we report the activities of a research project aimed at studying the electrification of a commercial 125cc motorcycle. The main focus of is the development of the controller of the Power Management System. Two strategies are proposed, a simpler one based on the evaluation of RPM derivative to select the boost/charge phase for the electric machine, and a more sophisticated *optimal control* strategy studied to obtain the best performance (in terms of available torque) while keeping the battery pack around its optimal operating point. Since the considered vehicle is a sports vehicle, the target is the utilization of the electric engine as a *torque-boost supplier*. Starting from an exiting software tool for simulation provided by VI-GRADE, VI-BIKEREALTIME (VI-Grade, 2012), a flexible SIMULINK environment has been developed. An easily customizable model for the accumulators has been derived, together with a map-based model for the electric engine. Simulations have been carried out using different configuration for the virtual rider tested on different tracks, elaborated with a dedicated tool to evaluate the performance of the control strategies among different driving conditions. The results show that satisfying performance can be obtained, in terms of both available torque utilization and battery management. Finally, the simpler of the two strategies have been implemented in the Electronic Control Unit (ECU) of a prototype and tested on track.

This part of the dissertation is organized as follows.

In **Chapter 2** the different elements of the prototype is described, namely the chosen motorbike, accumulators and electric machine. The work by Ferrari (2010) reports a dimensioning study that defines the selected components.

In **Chapter 3** the proposed control strategies are depicted. In particular, a modelization of the accumulators, based on the information from datasheets is provided, that will be useful for the control strategies proposed. Both the algorithms will be described in details.

In **Chapter 4** some results are illustrated. The SIMULINK -based virtual environment is described and simulations on a digital version of Silverstone (UK) racing track, as well as on a track with urban features are provided. The simulations takes advantage of the presence of a virtual rider, whose importance is crucial to close the loop and validate the proposed control strategies. On road tests are also reported, compared to simulations on the same test track (PIAGGIO test track in Pontedera, Italy). Greater importance will be given on the boost phase, since instruments for

measuring the charge of the batteries in real time were not available at the time of the test.

In **Chapter 5** final remarks will be given.

2

The Prototype

In this section the hybrid motorbike is presented, characterizing the basic vehicle and the components of the electric propulsion system.

As seen in the introduction, several configurations for coupling internal combustion and electric engines are available in the literature. The chosen configuration for the considered prototype is the parallel one, i.e. the two engines can run separately or together. The electric machine replaces the alternator and is keyed on the drive shaft, resulting in the constraint that both engines run at the same rounds per minute.

In the present study, the prototype is considered to be composed of three parts, the motorcycle, the electric engine, and the batteries.

2.1 Motorbike: Aprilia RS4 125

The commercial vehicle considered for the prototype is the RS4 125 (figure 2.1) produced by the motorcycle manufacturer Aprilia Aprilia S.p.A. (2012). The vehicle is powered by a new 125cc single-cylinder, 4-stroke, 4-valve, liquid cooled engine with electronic injection, adopting a double overhead camshaft (DOHC). This engine provides the maximum power allowed for its category, i.e. 11 kW. To ensure maximum exploitation of the power and a wide range of use, the engine is associated to a six-speed gearbox. Table 2.1 lists the specifications for the engine and transmission. The chassis is in aluminum, intended to support at best the engine performance while ensuring maximum driveability. Table 2.2 shows the main characteristics of the chassis.

Figure 2.1: Aprilia RS4 125.

Engine	Four stroke liquid cooled single cylinder with electronic injection and 4 valves
Bore *times* Stroke	58 × 47 mm
Total displacement	124.8 cc
Compression ratio	12,5 ± 0,5:1
Gearbox	6 ratios
Drive	Primary: gears, 69/29 Final: chain, 60/13
Emissions	Euro 3

Table 2.1: Technical sheet RS4 125: engine and transmission.

Chassis	Aluminium perimeter frame
Front suspension	41 mm upside down fork Wheel excursion 110 mm
Rear suspension	Asymmetric swingarm with monoshock Wheel travel 130 mm
Brakes	Front: 300 mm stainless steel disc with radial 4 piston calliper Rear: 218 mm stainless steel disc and calliper with single 30 mm piston
Wheels	In light alloy with 6 split spokes Ant.: $2.75 \times 1''$ Post.: $3.50 \times 17''$
Dimensions	Maximum length 1953 mm Maximum width 740 mm Maximum height at top fairing 1.138 mm Saddle height 820 mm Wheelbase 1.350 mm

Table 2.2: Technical sheet RS4 125: chiassis and sizes.

The exhaust system, fully integrated in the bottom hull, improves the dynamic behavior due to the centralization of the masses of the vehicle.

The characterization of the motorcycle is defined by *torque* and *power* curves, obtained on the test bench with *maximum throttle opening* condition: numeric measurements are in table 2.3, while figures 2.2 and 2.3 report the graphical representations as functions of RPM (Revolutions Per Minute)

RPM	Torque [Nm]	Power [kW]	Power [HP]
1000	1.52	0.16	0.22
1500	2.28	0.36	0.49
2000	3.04	0.64	0.87
2500	3.80	0.99	1.35
3000	4.56	1.43	1.95
3500	5.32	1.95	2.65
4000	6.08	2.55	3.46
4500	7.06	3.33	4.52
5000	7.50	3.93	5.34
5500	8.05	4.64	6.31
6000	8.31	5.22	7.10
6500	8.75	5.96	8.10
7000	10.05	7.37	10.02
7500	10.68	8.39	11.41
8000	10.73	8.99	12.23
8500	10.33	9.20	12.51
9000	10.44	9.84	13.38
9250	10.33	10.01	13.61
9500	10.15	10.10	13.73
9750	9.91	10.12	13.76
10000	9.68	10.14	13.79
10250	9.42	10.11	13.75
10500	8.96	9.85	13.40
10750	8.53	9.60	13.06
11000	8.04	9.26	12.60

Table 2.3: RS4 125 torque and power characterization.

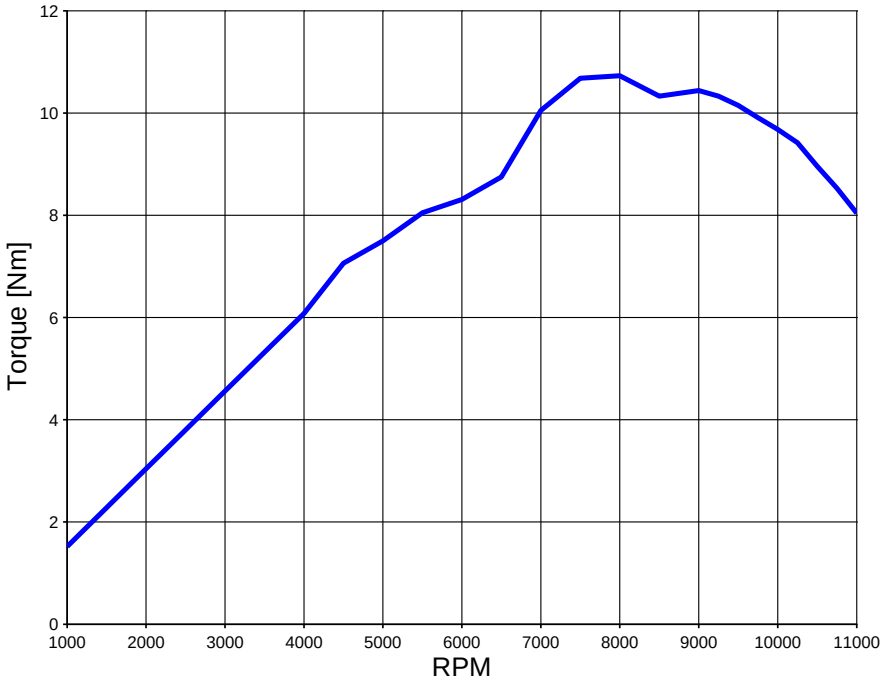


Figure 2.2: Torque characteristic of Aprilia RS4 125.

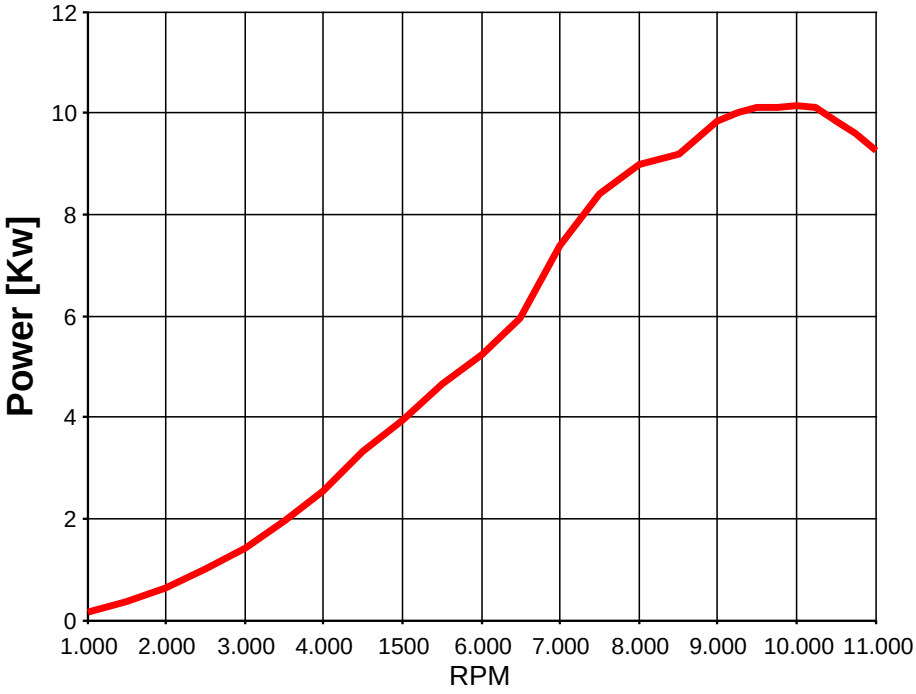


Figure 2.3: Power characteristic of Aprilia RS4 125.

2.2 Hybrid system

The electrical part of the hybrid system has been designed by the Department of Electrical Engineering of the University of Padova. Details on the design can be found in the work by Ferrari (2010) where the sizing of the batteries and the inverter, as well as the project of the electric motor has been carried out from scratch.

2.2.1 Batteries

The battery pack chosen for the application at hand has been designed considering *Lithium-Polymer* cells made by KOKAM (Kokam). The chosen model for the single cell is SLPB100216216H. Lithium - Polymer (Li-Po) technology offers many advantages over traditional Lithium-ion (Li-ion) cell including:

- elimination of the risk of fire, since the cells consist of non-flammable polymers;
- the flexible structure allows for a lighter case for the battery pack, thereby obtaining a considerable saving of weight. Their prismatic structure also permits a very dense packing, resulting in an increase of the energy density of approximately 20% compared to traditional Li-ion cells having a cylindrical shape. Figure 2.4 sketches the dimensions of a single cell.

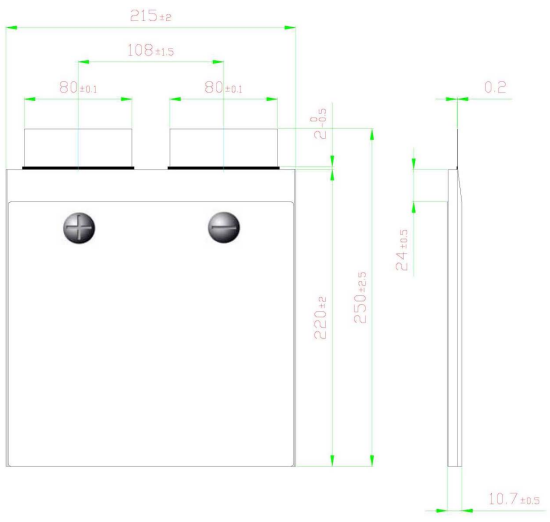


Figure 2.4: Accumulators KOKAM SLPB100216216H: sizing.

Each cell has the features reported in table 2.4. Of particular interest are the conditions of *charge* and *discharge*, which describe the voltage and maximum current in the two

phases, and the *cut-off* voltage of discharge. This value indicates the minimum voltage allowed for the discharging phase, in order to avoid permanent damage or remarkable reduction of the lifetime of the accumulator.

Capacity		40 Ah
Nominal voltage		3.7 V
Charge conditions	Max. current	80 A
	Voltage	4.2 ± 0.03 V
Discharge conditions	DC current	200 A
	Max. current	400 A
	Cut-off voltage	2.7 V
Working temperature	Charge	0 ∼ 40° C
	Discharge	-20 ∼ 60° C
Dimensions	Thickness	10.7 ± 0.5 mm
	Width	210 ± 2.0 mm
	Height	220 ± 2.0 mm
Weight		1100 ± 40 g

Table 2.4: Accumulators KOKAM SLPB100216216H: characteristics.

A preliminary analysis performed on the basis of static simulations suggests the use of a four cell battery pack (De Simoi, 2011); in figures 2.5 and 2.6 the charge and discharge curves from data-sheet are shown, for a single cell. These curves are fundamental to evaluate the performance of the accumulator during charging and discharging. In particular, exploiting these curves it is possible to obtain an estimate of the values of voltage as a function of the charging time and the State Of Charge (SOC) of the cell: the battery model integrated in the proposed control algorithms is based on the information derived from these curves.

For the application at hand, a crucial information about the batteries is the specification of the current value as a function of the *battery capacity*. In fact, as evident in figures 2.5–2.6, different curves are reported depending on a value, indicated as iC with $i = 0.5, 1, 2$ and so on. This corresponds to a constant charge/discharge current (in Ampere) of value equals to i times the nominal maximum capacity of the battery (expressed in Ampere per hour): i.e., if we consider the charging curve at 2C for the specific cells, it means a constant current value of 80 A, twice the capacity (that is 40 Ah). In the same way, a 0.5C discharge curve represents the behaviour in the discharge phase when the current is constant at a value of 20 A, half the capacity. This is a common convention in this

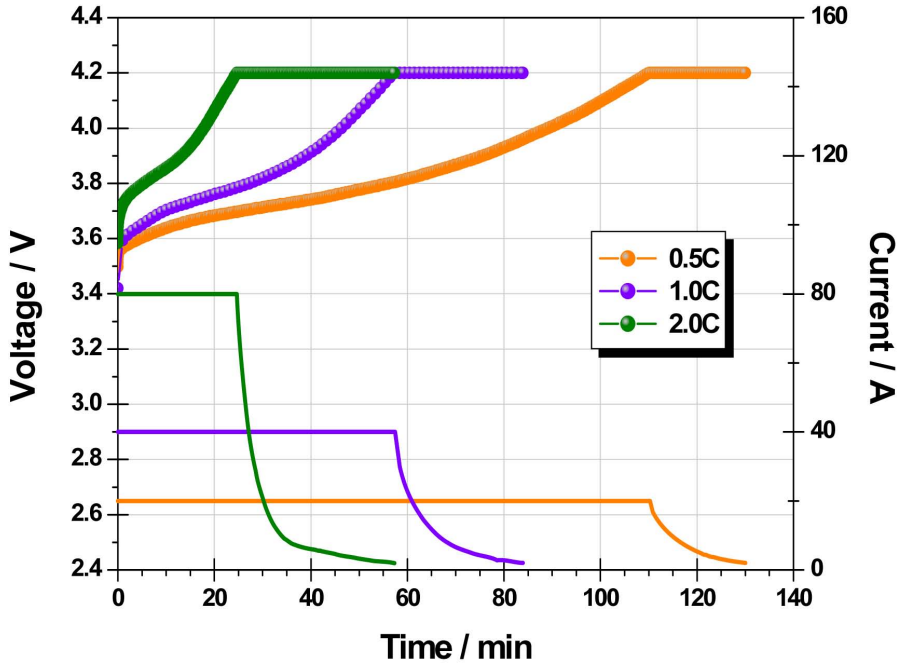


Figure 2.5: Accumulators KOKAM SLPB100216216H: charge curve.

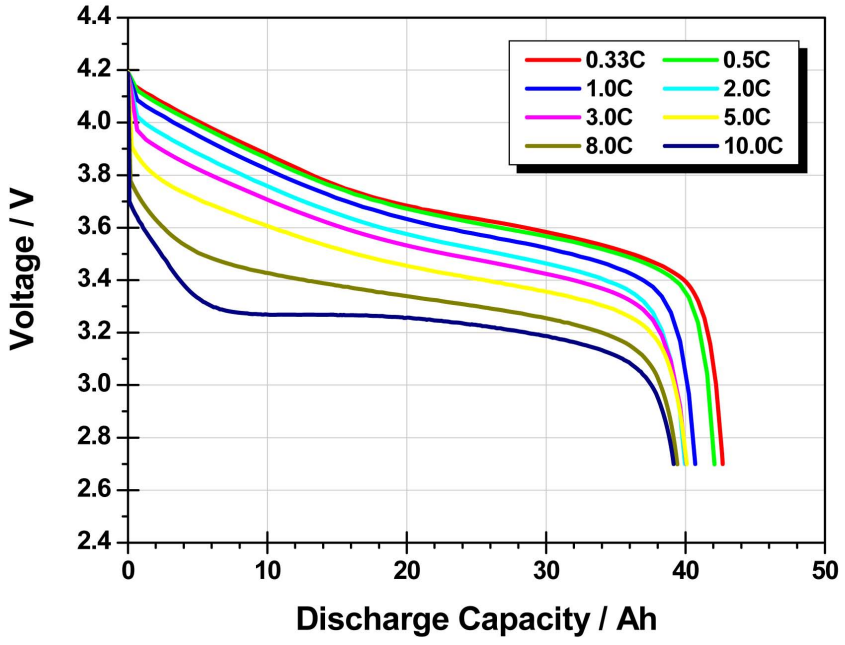


Figure 2.6: Accumulators KOKAM SLPB100216216H: discharge curve.

field, and it is important since the specification of the problem requires to consider this parameter.

It is worth noticing that, while the discharge curve represents the *voltage* as a function of the discharge capacity, i.e. of the residual stored energy, the charge one is a function of time and, for each value of the charging current, two phases are distinguished

1. constant current (equal to the imposed value, multiple of the capacity) and increasing voltage;

2. constant voltage (equal to its maximum value for the cell) and decreasing current to zero.

This will be another critical aspect to be considered in the modeling of the batteries.

2.2.2 Electric motor

Most of the alternators that equip commercial motorbikes are Surface Permanent Magnet (SPM) synchronous electrical machines: this solution is usually adopted to exploit the high torque density within the reduced weight and space requirements.

Following this idea, to make the best use of the available space the electric machine has an internal stator and a permanent magnet external rotor of type SPM: the device substitutes the native alternator (Ferrari, 2010), and is located in the same place as highlighted in figure 2.7. The maximum useful length is 62.2 mm, with a diameter of 135 mm; the weight is approximately 4.8 kilos. The chosen hybrid configuration is the *parallel* one, hence the electric motor is keyed on the same shaft as the internal combustion engine: it can be considered a mild hybrid with a BAS-like configuration.

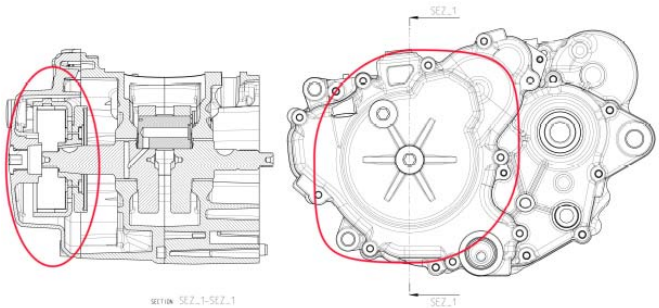


Figure 2.7: Electric machine location in the motorbike.

The designed motor is capable to supply torque such that the overall engine torque profile for the hybrid vehicle is the one shown in figure 2.8. Note that the highest torque requested from the electric motor is approximately 5 Nm. Table 2.5 shows the boost value provided for increasing RPM, evaluated at maximum throttle opening.

RPM	Electric torque [Nm]	Endothermic torque [Nm]	Hybrid torque [Nm]
1000	5.0	1.52	6.52
1500	4.8	2.28	7.08
2000	4.6	3.04	7.64
2500	4.4	3.80	8.1
3000	3.9	4.56	8.46
3500	3.4	5.32	8.72
4000	3.0	6.08	9.08
4500	2.3	7.06	9.36
5000	2.1	7.50	9.6
5500	1.9	8.05	9.95
6000	1.9	8.31	10.21
6500	1.7	8.75	10.45
7000	0.5	10.05	10.55
7500	0.0	10.68	10.68
8000	0.0	10.73	10.73
8500	0.0	10.33	10.33
9000	0.0	10.44	10.44
9250	0.0	10.33	10.33
9500	0.0	10.15	10.15
9750	0.0	9.91	9.91
10000	0.0	9.68	9.68
10250	0.0	9.42	9.42
10500	0.0	8.96	8.96
10750	0.0	8.53	8.53
11000	0.0	8.04	8.04

Table 2.5: Torque performance of the electric machine, the endothermic engine and the hybrid motor.

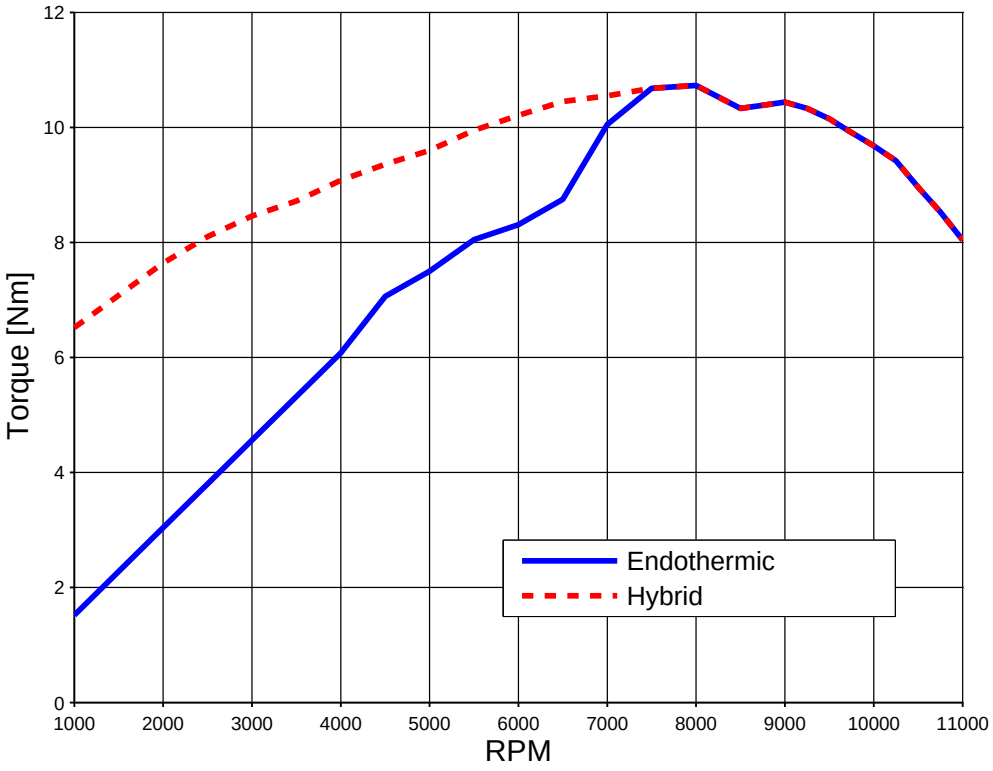


Figure 2.8: Torque characteristic of the hybrid engine.

Remark 2.2.1. The electric machine requires a power electronics system too, to interface with the battery pack. This system is set between the motor and the accumulators, and is constituted by the cascade of two components, namely

1. a *power inverter*, whose aim is to convert the DC current from the accumulators into AC current for the engine, when it is providing propulsion, and viceversa when it works as a generator and needs to transfer energy to the batteries via DC current;

2. a *DC/DC converter*, interposed between the inverter and the battery pack, to regularize the continuous current from the inverter to a proper value for the batteries.

A scheme is reported in figure 2.9. These components are necessary to the practical operation of the electric machine, but from a control point of view it is not important to include them in the model, at least for the purpose of this dissertation: in fact, the goal is the management of the power system in order to obtain a proper integration of the boost from the rider point of view, and not to control each single component of the loop. In this sense, the power electronics group will not be considered in the development, and will supposed only to provide the commands from and to the electric motor, with respect

to the accumulators; the only parameter that to be considered is its efficiency η, that is estimated as to have and approximate value of 0.9 (Ferrari, 2010).

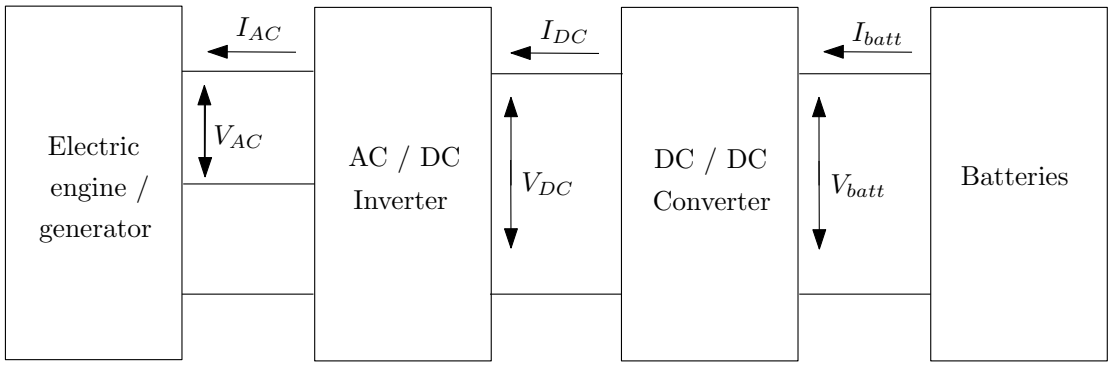


Figure 2.9: Complete scheme of the electric machine.

3

Control Strategies

The hybridisation of a vehicle is a challenging task, first of all for its physical design. Combining a brand new propulsion system with all its parts including engine, source of power, stabilizing controllers and other regulators together with an already existing vehicle is not immediate, and require a thorough analysis to exploit the available space and integrate the new elements without undermining the physical characteristics, e.g. masses distribution, gravity center position etc. that can have an impact on the usability of the vehicle. Moreover, the realization of an electric machine combined with a battery pack system is per se a challenging task, and requires a careful control system design. The *Power Management System* (PMS) is a critical point of such devices, since the batteries management is not trivial and requires the analysis of the features of the accumulators, their behaviour at varying voltage and current, temperature and other physical quantities that have to be taken care of. This aspect affects the overall performance of the hybrid system, in term safety, reusability (a bad management is likely to damage the capacity of

the batteries), and further expensive maintenance. This management has to be integrated in the overall control framework for the hybrid vehicle, that has the task to couple the two source of motion in order to obtain the final, desired goal, namely emission reduction, torque and speed improvement, money saving or a combination of such (and possibly other) aspects. All of this has obviously to be realised in such a way to be transparent to the final user, so that the coupling of the two system results in a safe and pleasant driving experience rather than in the necessity of learning a new way to handle the vehicle.

It is trivial to figure out as this challenge is way harder for two-wheels vehicles: the available space is reduced with respect to cars, an increase in the weight and volumes is a more delicate issue (given the higher degrees of freedom such vehicles are subject to) and the impact on the driver is much more relevant, related to safety and driveability issues. This is even more critical when dealing with sport motorbikes, as the one considered in the design, rather than with mopeds: the expectations from the user is different and requires more attention from the performance point of view.

In this chapter, a control system that combines a PMS and the regulation of torque coupling is proposed. In particular, two different strategies will be shown, with different degrees of complexity. The first one looks to a fast implementation and is based on a simpler idea, more conservative with regard to battery exploitation; nevertheless, it integrates adaptive features for a smarter usage of power. The second one is an optimal control strategy which aims at the regulation of stored power by managing in a more effective way the integration between the two powertrains.

Upstream, a common element is the selection of the operating mode of the electric machine, as a motor or a a generator: the algorithm that performs this operation is based on a model of the accumulators, that will be illustrated in the following section.

3.1 Battery model

Deriving a model for a battery is a challenging issue, since its behaviour depends on a wide variety of internal and external variables, e.g. internal temperature, *ageing* (number of charge and discharge cycles), operational conditions, and many more. Variations in these quantities result in modified internal resistance, capacity reduction and, more in general, accumulator damage. Some kinds of batteries (e.g. nickel-cadmium, nickel-metal hydride) suffer of the *memory effect*, that occurs in the event that they are repeatedly recharged before the complete exhaustion of the available energy, causing alterations of the internal crystal structures that reduce the storage capacity of a quantity equal to the residual (“memorized”) energy at the beginning of the cycle charge. In the hybrid

vehicles context, having a careful model for batteries is crucial, not only in the design stage but also for control purposes, in order to estimate meaningful information on the battery state (as State Of Charge, i.e. the normalized quantity of available energy, or temperature) from easily measurable quantities (as voltage or current). On the other hand, a complete modeling of the battery is too burdensome, so the representations are adapted to capture relevant aspects for this kind of studies.

The most common models for batteries are

- *electrochemical* models, which exploit the physical laws that regulate the electrochemical phenomena inside the battery (Debert, Colin, Mensler, Chamaillard, and Guzzella, 2008; Smith, 2010; Smith, Rahn, and Wang, 2010; Speltino, Di Domenico, Fiengo, and Stefanopoulou, 2009). Such models are very precise, but complex and the knowledge of a great number of parameters as well as availability of sophisticated analysis tools are needed for their implementation. Moreover, they have heavy computational burden and make real-time simulation and application quite difficult. An insight on this approach can be found in the work by Marangon (2010), where an Extended Kalman Filter-based algorithm is proposed;

- *electric* models, which exploit the *equivalent circuit* and its laws to represent the electrical behaviour of the battery (Chen and Rincon-Mora, 2006). Despite being quite an empirical approach, it allows a simple modeling of the accumulators dynamics that can be improved if needed, starting from the measurements of interest, by increasing or decreasing the order of the system and highlighting the trends of voltage and current. The reference circuit diagram is the classic RC circuit. To improve the accuracy, the values of resistors and capacitors can be expressed as a function of temperature, State Of Charge (SOC) and current value.

The power management system has to provide the status of the batteries (temperature, state of charge, voltage at the poles) while ensuring that they are used properly, avoiding damage. Taking advantage of a correct modeling of the battery PMS performs the following activities:

1. *direct measurement* of the main physical features of the battery pack as temperature, voltage and current of the single cell or the entire pack;

2. *estimation* of the information not directly available, as *State Of Charge* (SOC) (Li, Anderson, Song, Phillips, and Wang, 2011);

3. *communication* of the values obtained from previous steps;

4. *control* of charge and discharge current as well as the cooling system of the battery, to avoid damage and unstable conditions;

5. *balance* of the state of charge of each cell in order to increase the lifetime of the battery. The SOC of individual cells may in fact differ between each other due to small construction differences, resulting in a reduction of the total capacity.

All these factors have been considered in the development of the model of the accumulators for the application at hand. The real-time purposes ask for a model capable to catch all the fundamental dynamics in order to obtain reliable estimations of the values of interest (SOC, voltage V, current I); on the other hand, a test bench for measuring the parameters of the real battery pack has not been made available. For this reasons, a simple electric circuit as been adopted (Figure 3.1), considering the battery as a *real voltage source* controlled by SOC, hence the supplied voltage is a function $V_{batt}(SOC)$; the same holds for the current $I_{batt}(SOC)$. The proposed modeling method is inspired by

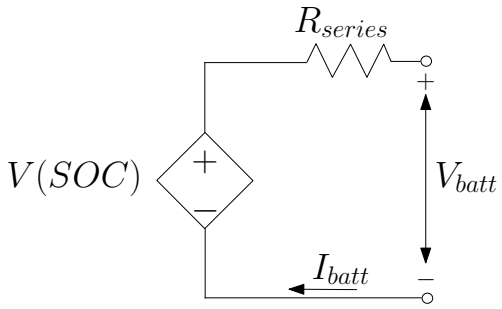


Figure 3.1: Equivalent electric circuit considered for the battery modeling.

Yurkovich, Guezennec, and Bornatico (2008) and relies on information directly available from the battery pack data-sheet. Point-wise charge and discharge maps are derived from the characteristic curves included in the previous chapter, in figures 2.5 – 2.6, and applying *polynomial fitting*, analytical representations for $V_{batt}(SOC)$, $I_{batt}(SOC)$ are calculated. Note that

1. characteristic curves are *not* directly given as function of SOC, hence preliminary calculations have to be performed to obtain the functions of interest;

2. the model assumes negligible temperature variations (which can be included if information from the data-sheet are available (Dougal, 2002)) and charge losses due to chemical reactions. For the purpose of prototyping control design and simulation, and without a better characterization of the accumulators, these hypotheses are acceptable.

Once V_{batt} and I_{batt} are available in such fashion, the *power* (and energy) supplied by the accumulators are easily computable.

It is worth remarking again the fact that in these datasheets, both charge and discharge currents are considered constant at values multiple of the nominal capacity, i.e. $I_{batt} = kC$ A, with $k = 0.5, 1, 2, \ldots$ and $C = 40$. The project specifications require to consider this value as a constant, hence this modeling procedure is coherent.

3.1.1 Charge functions

In figure 2.5, charge voltage and current are given as functions of time, $v^C_{batt}(t)$ and $i^C_{batt}(t)$, for different charging strategies (namely, different constant value for the charge current during the voltage transient). After the calculation of $v^C_{batt}(t)$ and $i^C_{batt}(t)$ as polynomial fitting of sampled values of the plot, to make these functions function of SOC, the *charge power* function $P^C_{batt}(t)$ is calculated as

$$P^C_{batt}(t) = v^C_{batt}(t) \cdot i^C_{batt}(t) \tag{3.1}$$

and then integrated to obtain the available energy

$$E^C_{batt}(t) = \int_0^t (P^C_{batt}(t) - V_0 \cdot i^C_{batt}(t))\,\mathrm{d}t, \tag{3.2}$$

where V_0 is the *cut-off* voltage value (i.e. the voltage lower bound to avoid damages, given in the data-sheet). By normalizing $E^C_{batt}(t)$ with respect to its maximum value, $SOC(t)$ is computed. Finally, having described each quantity as a function of time, it's easy to compose the functions to obtain $V^C_{batt}(SOC)$ and $I^C_{batt}(SOC)$.

Note from figure 2.5 that the charge characteristic is divided in two phases. At first, for $SOC \to 0$, $I^C_{batt}(SOC)$ is constant and equal to I_{kC}, while $V^C_{batt}(SOC)$ increases starting the minimum value V_0. When the voltage reaches its maximum value $V_{\max}$, the current has a decreasing transient and becomes null when $SOC = 100\%$.

This last transient is not automatically achieved by the cells: the current comes from the electric machine, and if it is not limited when the voltage has reached its maximum, the accumulators will keep on storing the energy until the maximum capacity is reached. If the current continues to be provided, the consequent generated energy will be converted into heat and dissipated, increasing the temperature and risking a permanent damage for the pack, as well as safety issues for the user. For this reasons, the accumulators usually come with a *charge controller* that has the capabilities of limit the absorbed current when the SOC crosses a specific value. If the batteries are not provided with such system, it

has to be integrated within the PMS. The proposed modelization takes into account this feature, hence $V^C_{batt}(SOC)$ and $I^C_{batt}(SOC)$ are defined as

$$V^C_{batt}(SOC) = \begin{cases} f_{V^C_{batt}}(SOC) & \text{if } SOC < SOC_{\text{lim}} \\ V_{\text{max}} & \text{if } SOC \geq SOC_{\text{lim}} \end{cases} \tag{3.3}$$

$$I^C_{batt}(SOC) = \begin{cases} I_{kC} & \text{if } SOC < SOC_{\text{lim}} \\ f_{I^C_{batt}}(SOC) & \text{if } SOC \geq SOC_{\text{lim}} \end{cases} \tag{3.4}$$

where $f_{V^C_{batt}}(SOC)$, $f_{I^C_{batt}}(SOC)$ are polynomial functions of appropriate degree obtained by fitting the transient data (Dougal, 2002),

$$f_{V^C_{batt}}(SOC) = \sum_{k=0}^{n} v_k \cdot SOC^k, \tag{3.5}$$

$$f_{I^C_{batt}}(SOC) = \sum_{k=0}^{m} i_k \cdot SOC^k. \tag{3.6}$$

3.1.2 Discharge functions

The discharge law is much easier to derive in our case, since it is directly obtained from the characteristic curve (figure 2.6): provided that constant current I_{kC} is applied to the load (in this case, the electric motor), the voltage is given as a function of the *discharge capacity*, hence by polynomial fitting we obtain the function

$$f_{V^D_{batt}}(SOD) = \sum_{k=0}^{n} v_k \cdot SOD^k, \tag{3.7}$$

where SOD is the State Of Discharge, related to SOC as $SOD = 1 - SOC$. Hence, we obtain

$$V^D_{batt}(SOC) = f_{V^D_{batt}}(1 - SOC) \tag{3.8}$$

$$I^D_{batt}(SOC) = I_{kC}. \tag{3.9}$$

3.1.3 Power estimation

For the control and simulation purposes, the value of the SOC needs to be known at each time. Since it directly depends on the energy E, hence on the power P, it is useful to

introduce an estimation of the power supplied by and to the accumulators, depending on the operational state of the electric machine (as powertrain or generator, respectively).

In the boosting phase this estimation is simple, and is given by

$$P^D_{batt}(t) = \frac{\tau_{em}(t) \cdot \mathrm{RPM}(t)}{\eta_{em}}, \tag{3.10}$$

where t is the present instant, $\mathrm{RPM}(t)$ is the actual speed of the hybrid engine, and $\tau_{em}(t)$ the corresponding provided boost torque (that depends both on $\mathrm{RPM}(t)$ and the throttle opening). η_{em} is the total *efficiency* of the electric machine, comprising the motor, inverter and converter: to achieve the desired boost, the power from the accumulators have to compensate also for the physical losses. The spent energy (hence $SOC(t)$) is then computed by integrating $P^D_{batt}(t)$.

When the electric machine operates as a generator, it is necessary to have a real-time estimate of the instantaneous power absorbed by the batteries, $P^C_{batt}(t)$, so that $SOC(t)$ can be calculated and it is possible to model the *charge controller* described in section 3.1.1. Based on an approach similar to the one described in that section, the model describes $P^C_{batt}(t)$ as a polynomial function calculated with (3.1): for different values of t, correspondent points $v^C_{batt}(t)$ and $i^C_{batt}(t)$ are extrapolated from the data-sheets, giving a set of points for $P^C_{batt}(t)$ and the corresponding function is calculated via polynomial fitting as

$$P^C_{batt}(t) = \sum_{k=0}^{n} p_k \cdot t^k. \tag{3.11}$$

Now, the energy of the battery $E(t)$ is an increasing monotonic function, being the integral of the power which assumes only non-negative values, hence $SOC(t) = E(t)/E_{max}$ is monotonic as well. If we define SOC_{lim} as the value after which the current transient starts, we have that for $SOC(t) < SOC_{\mathrm{lim}}$ the function is monotonic, as well as for $SOC(t) \geq SOC_{\mathrm{lim}}$, hence it is possible to look at the inverse map to get the time instant $\bar{t}$ relative to a value of P^C_{batt}: being T_s the sample time, the estimation of the charging power is computed from the polynomial model as $P^C_{batt}(\bar{t} + T_s)$. The estimate is then integrated to obtain $E(t)$ and the correspondent $SOC(t)$.

3.2 Electric machine: selection of the operation mode

The control system for the hybrid motorcycle has to assure the best use of the electrical machine in terms of *supplied torque boost*, while avoiding an excessive consumption of the battery in order to prevent damages and possible safety issues. The electric motor works

as a generator during charge steps and as an engine during boost: the controller has to select which is correct operation mode, depending on the actual telemetry information.

As said in section 3.1.1, when the battery voltage is less than $V_{\max}$, the current supplied by the generator is constant, and can be set as a multiple of the nominal capacity. To each value of I_{kC} corresponds a lower bound on $\mathrm{RPM}(t)$, $\overline{\mathrm{RPM}}_{kC}$, below which the generator cannot supply enough power to provide the requested current. Since charging batteries is a slower but safer task than discharging, this operation mode must be activated whenever it's possible, i.e. when the motorcycle is at constant or decreasing speed and $\mathrm{RPM}(t) \geq \overline{\mathrm{RPM}}_{kC}$: hence, the key point is to decide whether the present vehicle set-up allows the motor to operate as a generator. Defined the motorcycle speed as $v(t)$, the gear signal as $g(t)$ and the energy level as $E(t)$, four conditions must hold:

1. $\dot{v}(t) \leq \delta v_0$, where $\delta v_0 > 0$ is a threshold below which the speed can be considered constant or decreasing. It is set as greater than zero to avoid inconsistencies due to the numerical calculation of the derivative;

2. $\mathrm{RPM}(t) \geq \overline{\mathrm{RPM}}_{kC}$;

3. $\dot{g}(t) \leq 0$, to avoid problems on negative $\dot{v}(t)$ values due to gear-shift and not to real speed variations;

4. $E(t) \leq E_{\max}$: the motor acts as generator only if accumulators are not full-charged.

Note that the third condition has to be temporized, since the variations on $g(t)$ are clearly faster than the correspondent variations on $v(t)$.

The set of conditions that allow the electrical machine to act as an engine, providing torque contribution, are derived in a very similar fashion to those presented for the charge phase, in particular

1. $\dot{v}(t) > \delta v_0$: boost is applied during accelerations;

2. $\overline{\mathrm{RPM}}_L < \mathrm{RPM}(t) < \overline{\mathrm{RPM}}_H$: the boost acts within a RPM range specified at design stage byt setting the two thresholds;

3. $\dot{g}(t) < 0$: the condition on gear shift holds in the same way;

4. $E(t) > E_{\min}$: boost is applied only if energy is available in the accumulators.

3.3 Charge control

The charging phase is, in some way, an easier task for the control system: in fact when the electric machine acts as a generator, thanks to the parallel architecture considered it does not affect the driving behaviour of the vehicle, hence the controller has to operate as PMS only. Nevertheless, in this stage the criticalities are related to the slow dynamics of the accumulators: charge is always a slower task than discharge, and this can be seen by their characteristics in table 2.4 in chapter 2, where the maximum current provided during discharge is five times larger than the maximum charge current. In this sense, it is crucial to elaborate a strategy that exploits at best the periods when the charge is allowed.

The strategy in section 3.2 highlights the fact that the charge has to take place whenever possible if the motorbike is not accelerating, with the limitation of a minimum RPM value. Below this threshold, the electric machine is not able to generate enough power to provide the desired current. As already stated, this current is constant until the voltage of each cell of the battery pack has reached its maximum value $V_{\max}$, than the charge regulator forces it to decrease: hence, the critical phase is for voltage lower than $V_{\max}$, when the highest value for the current has to be provided. The chosen cells allow charge currents up to 2C, and reasonable values for the application are 1C and 2C. In the design of the motor, Ferrari (2010) computes the values of the two thresholds as

$$\overline{\mathrm{RPM}}_{1C} = 2500 \tag{3.12}$$

$$\overline{\mathrm{RPM}}_{2C} = 5000 \tag{3.13}$$

The higher is the charging mode, one the one hand the faster is energy accumulation, on the other hand the higher is $\overline{\mathrm{RPM}}_{kC}$. Depending on the driving context (route, driver's behaviour, traffic), the $\overline{\mathrm{RPM}}_{2C}$ constraint might be too high to allow a satisfactory recharge. The proposed charge control algorithm aims to extend the exploitation of the charging time by operating an automatic switch of charging modes between 1C and 2C, based on recent driving conditions. In this way the generator capabilities can be used at best, allowing to provide the most affordable current value during the wider possible time.

Once a time observation window T_{obs} has been set, the time intervals with $\mathrm{RPM}(t) > \overline{\mathrm{RPM}}_{2C}$ and $\overline{\mathrm{RPM}}_{1C} < \mathrm{RPM}(t) < \overline{\mathrm{RPM}}_{2C}$, t_{2C} and t_{1C} respectively, are computed and compared, and the corresponding charging mode is then applied. An hysteresis-based strategy on the difference between the two time values is applied to avoid chattering of the switching action. The procedure is summarized in algorithm 1. This strategy is effective under the hypothesis that the driving style is kept similar during a certain amount

of time, as it usually happens; note that the behaviour can be regulated by adjusting the threshold ε for the hysteresis as well as the observation window T_{obs}.

Algorithm 1 Charge management - charge current switching

Obtain current time t and RPM(t)
if charge conditions hold **then**
  $t_{obs} \leftarrow t_{obs} + T_s$ {t_{obs}: time elapsed during the current evaluation; T_s: sampling step}
  if $\overline{\text{RPM}}_{1C} < \text{RPM}(t) < \overline{\text{RPM}}_{2C}$ **then**
    $t_{1C} \leftarrow t_{1C} + T_s$
  else
    if $\text{RPM}(t) \geq \overline{\text{RPM}}_{2C}$ **then**
      $t_{2C} \leftarrow t_{2C} + T_s$
    end if
  end if
  if $t_{obs} = T_{obs}$ **then**
    if $t_{1C} > t_{2C}$ **and** $|(t_{1C} - t_{2C})| > \varepsilon$ **then**
      set 1C charging mode
    else
      set 2C charging mode
    end if
  end if
end if

3.4 Boost control: Adaptive Boost

The boost stage is more critical than the charge stage, from the point of view of both vehicle stability and battery management. In fact, when the batteries are required to supply energy, the discharge dynamics are way faster than charge ones: in table 2.4 the specifications show that a single cell is capable of providing a current up to 400 A, five times higher than the maximum absorbable one. More, during boost the electrical machine behaves as an engine, and contributes to the motion with and additive torque which directly affects the dynamic behaviour of the motorbike. This last aspect has to be regarded with particular care, since an unusual feedback from the vehicle can result in a major safety issue for the rider.

The proposed strategy is based on a quite simple heuristics, elaborated to be quickly implemented in the Electronic Control Unit (ECU) of the prototype, although it contemplates useful adaptive features. The reference torque map of the electrical engine (reported in the previous chapter, table 2.5) is a *full-boost* map: for each RPM value, it includes the *maximum* torque values that the motor can provide, meaning the highest

electrical power request to the accumulators. For the same RPM value, a lower torque request leads to lower power consumption. Hence, the idea is to introduce a *scaling factor* σ to reduce the request of electric torque, proportional to the current SOC. With this respect, the *adaptation* is performed on the residual energy of the battery pack, aiming to avoid excessive stress when it reaches low value.

The control variable for this strategy is the SOC itself. Two thresholds are set for the SOC, an upper one, $\overline{SOC}_U$ and a lower one, $\overline{SOC}_L$. If $SOC \geq \overline{SOC}_U$, full-boost is applied, if $SOC \leq \overline{SOC}_L$ boost is not allowed. For intermediate values of SOC, the scaling factor σ is computed as

$$\sigma = \frac{SOC - \overline{SOC}_L}{\overline{SOC}_U - \overline{SOC}_L}. \tag{3.14}$$

Note that σ varies linearly, but this behaviour can be easily modified for a more sophisticated one.

The described algorithm may suffer of *chattering* around the thresholds. Suppose that SOC is near the lower threshold and the route or traffic conditions force the rider to behave in a way that does not allow for durable charge: the SOC is likely to vary its value by repeatedly increasing and decreasing. The strategy described so far will enter and exit the boost state as well, a stressful behaviour for both the electric machine and the mechanical components of the powertrain. To prevent this condition, an hysteresis is introduce on the thresholds, based on the evaluation of the SOC trend. If the derivative show and increase, both the thresholds are increased by a factor ε, viceversa with negative derivative they are decreased by the same factor. ε has to be tuned according to the expected usage of the motorbike. The procedure is summarized in algorithm 2.

On the same fashion, to avoid safety issues for an abrupt addition of torque to the vehicle that can make the rider lose control, there is an upper layer that forces a linear increase of the scaling factor from zero to the computed value, when the boost conditions hold and the contribution is provided, so to assure continuity in the torque supply.

Algorithm 2 Adaptive Boost Control: battery management

Obtain current $SOC(t)$ and RPM(t)
if boost conditions hold **then**
  obtain $\tau_{ice}^{\max}$ from electric torque map {$\tau_{ice}^{\max}$: maximum electric torque that can be supplied with RPM(t)}
  compute $\dot{SOC}(t)$
  if $\dot{SOC}(t) > 0$ **then**
    $SOC_U \leftarrow \overline{SOC}_U + \varepsilon$ {hysteresis update}
    $SOC_L \leftarrow \overline{SOC}_L + \varepsilon$
  else
    $SOC_U \leftarrow \overline{SOC}_U - \varepsilon$
    $SOC_L \leftarrow \overline{SOC}_L - \varepsilon$
  end if
  if $SOC(t) > SOC_U$ **then**
    $\sigma \leftarrow 1$
  else
    if $SOC_L < SOC(t) < SOC_U$ **then**
      $\sigma \leftarrow \frac{SOC(t) - SOC_L}{SOC_U - SOC_L}$
    end if
  else
    $\sigma \leftarrow 0$
  end if
  $\tau_{ice}(t) \leftarrow \sigma \cdot \tau_{ice}^{\max}$ {$\tau_{ice}(t)$: electric torque provided}
end if

3.5 Boost control: Adaptive Torque Splitter

This approach is based on an *optimal control* algorithm. This class of controllers are observed with increasing interest in the hybrid vehicles field. However, most of the approaches based on optimization methods, whether being static, numeric (dynamic programming) or analytical, requires a-priori knowledge of the driving cycle (speed profile, rider requests, etc.) to calculate the instantaneous values needed to achieve the objective of optimality at the end of the path. In the everyday use this is not realistic, hence the development of real-time optimization strategies is required so that, by analyzing the behaviour of the vehicle, at least sub-optimal performance are assured. To our purposes, of particular interest are the strategies whose aim is to calculate the *split* of *power* or *torque* that each engine must provide in order to minimize fuel consumption at the end of the driving cycle. By operating in real-time, the only available information are the data acquired from the analysis of recent telemetries, the current state of the vehicle and (possibly) a static set of values: then, by appropriate processing, the values of the best

energy flows are computed and applied.

Some of the approaches that are looked at with interest in the scientific community are

- *Adaptive Equivalent Consumption Strategies* (A-ECMS) (Musardo, Rizzoni, and Staccia, 2005) that extends the general ECMS framework by introducing an update of the parameters dependent on driving conditions. The general methodology, exploiting a particular function which allows to calculate an equivalence factor between the fuel and electric energy, computes at each instant the power values that both powertrains must supply in order to minimize the cost function, which represents the *equivalent fuel consumption* of a single, endothermic engine having the same torque and power profile of the real, hybrid one. The main weakness of this approach is the strong dependence of the equivalence factor from the particular driving cycle. A-ECMS tries to overcome this problematic through the use of a particular algorithm that, exploiting data from appropriate instruments (GPS), periodically adapts the equivalence factor according to the estimated load required by the path;

- *Model Predictive Control* can be applied considering the model of the system, with constraints imposed on the energy level of the battery. The prediction is limited to a reasonable interval for estimation of quantities of interest. The formulation by Koot, Kessels, DeJager, Heemels, VandenBosch, and Steinbuch (2005) minimizes a cost function representing the fuel consumed by the combustion engine; by exploiting a particular reformulation of the optimum problem, the prediction window is reduced to a single sampling instant, directly computable using the system model. Kessels, Koot, de Jager, van den Bosch, Aneke, and Kok (2007) extend the optimization to the entire power network of the vehicle, by means of a complex cost function which considers the efficiency of the battery and the electric drive. A recent strategy, presented by Fu, Ozguner, Tulpule, and Marano (2011), makes use of MPC to calculate the values of the energy flows that allow to minimize the consumption of fuel in the interval of prediction, by introducing constraints regarding the efficiency of the engine and the equality (within a tolerance range) of the SOC at the beginning and end of the time interval. In this case the prediction of the speed values required for the calculation of the cost function is carried out by exploiting different portions of existing telemetry;

- *Optimal Stochastic Control* exploits the stochastic modeling of one or more variables of interest to design the estimators that, based on current measurements, allow to

calculate the values at the next step and to apply the appropriate optimal control. This approach is proposed by Moura, Fathy, Callaway, and Stein (2011), with a stochastic modeling of the driving cycle that allows estimation of the required power; an optimization procedure is applied to obtain the power split that minimizes a cost function based on equivalent cost of fuel consumed by the combustion engine and the electric machine.

Following the "splitting" philosophy, we propose an algorithm based on the idea of *torque split*. This approach has been chosen because of the technical features of the prototype, that is, having the constraint that the two engines run at the same RPM value, it is convenient to regulate the torque provided by each engine. Considering that the final state of charge has to be maintained as close as possible to its maximum value, the algorithm focuses on the minimization of the use of the Internal Combustion Engine (ICE) during acceleration, while ensuring the autonomy of the battery to be preserved. To assure real time performance, the strategy has a *greedy* approach, and exploits an optimization based on a simple, torque-map-based model; the cost function is continuously adapted to the driving conditions.

3.5.1 Constraints and cost function

From the torque-maps, for each value of RPM(t) a correspondent minimum and maximum value for the endothermic torque $\tau_{ice}(t)$ and the electric one $\tau_{em}(t)$ are available, hence it is possible to derive the torque profile for the hybrid powertrain $\tau_r(t) = \tau_{ice}(t) + \tau_{em}(t)$. The problem is formulated so that, given the total torque request, the optimal *splitting* between the torque that can be provided by the electric motor and the internal combustion engine is computed, thus obtaining the best trade-off between battery consumption and fuel economy.

The constraints for the system are

$$\begin{cases} SOC_{\min} \leq SOC(t) \leq SOC_{\max} \\ \tau_r(t) = \tau_{em}(t) + \tau_{ice}(t) \\ \tau_{em}^{\min}(\text{RPM}(t)) \leq \tau_{em}(t) \leq \tau_{em}^{\max}(\text{RPM}(t)) \\ \tau_{ice}^{\min}(\text{RPM}(t)) \leq \tau_{ice}(t) \leq \tau_{ice}^{\max}(\text{RPM}(t)) \end{cases} \tag{3.15}$$

where the bounds on $SOC(t)$ are given to prevent battery damage (and usually $SOC_{\max} = 1$, while $SOC_{\max} > 0$ to avoid problems due to an excess of discharge), and the bounds on $\tau_{em}(t)$ and $\tau_{ice}(t)$ reflect the minimum and maximum torque supplied by the two

engines at a fixed RPM value (depending on the absorbed current and the throttle opening, respectively). Note that the constraints, depending on RPM(t), are *time-varying*.

The chosen cost function is quite simple, being a weighted sum of $\tau_{em}(t)$ and $\tau_{ice}(t)$,

$$J(\tau_{em}(t), \tau_{ice}(t)) = w_{em}(SOC(t), SOC_{\text{ref}}) \cdot \tau_{em}(t) + w_{ice}(SOC(t), SOC_{\text{ref}}) \cdot \tau_{ice}(t). \quad (3.16)$$

The weights are function of the current $SOC(t)$ and a reference value, SOC_{ref}, below which the use of the electric motor has to be penalized. The *adaptive* aspect of the control algorithm proposed involves the *dynamic adaptation* of w_{em} and w_{ice} depending on the state of the batteries (equation (3.17)), and of SOC_{ref} depending on recent driving conditions: the first are regulated to keep $SOC(t)$ around its optimal operating point SOC_{ref}, the latter varies according to the recent trend of batteries utilization (i.e. if boost has not been much provided SOC_{ref} is lowered and vice versa).

$$\begin{cases} w_{em}(SOC, SOC_{\text{ref}}) = w_{ice}(SOC, SOC_{\text{ref}}) & \text{if } SOC = SOC_{\text{ref}} \\ w_{em}(SOC, SOC_{\text{ref}}) < w_{ice}(SOC, SOC_{\text{ref}}) & \text{if } SOC > SOC_{\text{ref}} \\ w_{em}(SOC, SOC_{\text{ref}}) > w_{ice}(SOC, SOC_{\text{ref}}) & \text{if } SOC < SOC_{\text{ref}} \end{cases} \quad (3.17)$$

A *smoothing constraint* on the derivative $S\dot{O}C(t)$ has been introduced to avoid sudden variations on τ_{ice} request, which can lead to bad driving feelings and issues on driver's safety. Hence we have

$$S\dot{O}C(t) < \varphi(SOC(t), SOC_{\text{ref}}) \quad (3.18)$$

with $\phi(\cdot,\cdot)$ a function such that

$$\lim_{SOC \to SOC_{\text{ref}}} \varphi(SOC(t), SOC_{\text{ref}}) = 0. \quad (3.19)$$

3.5.2 Optimization

$J(\tau_{em}(t), \tau_{ice}(t))$ can be rewritten as a function of just $\tau_{ice}(t)$, hence becoming a single-variable optimization problem; the constraints can also be reformulated to depend on $\tau_{ice}(t)$ only and incorporated in $J(\tau_{ice}(t))$, as shown by De Simoi (2011). The control problem becomes a *single variable, unconstrained* optimization problem, whose resolution can be obtained via different methods. In our case, the cost function depends on time varying quantities, and this can yield to the presence of cusps with consequent numerical issues in the case the solver requires to derive that function. The chosen solver, based on the *golden-ratio method* (Press, Teukolsky, Vetterling, and Flannery, 2007), has been preferred for its resolution speed and the capability of dealing with non continuous cost

function. Being the controlled variable τ_{ice}, the actuator is the throttle, a viable solution since ride-by-wire technology is now popular on motorbikes (see for instance the work by Beghi, Nardo, and Stevanato (2006)) and is integrated within the prototype.

4

Simulation and On-Track Results

The research described in this part of the dissertation has been carried on in the framework of a project in partnership with motorbikes manufacturer APRILIA S.P.A., the Department of Electrical Engineering of the University of Padova and other industrial partners, and it has already been stated that the final goal was the realization of a working hybrid prototype of the 125cc motorcycle RS4 125. To this respect, the validation of the proposed control and battery management strategies has been set as a necessary step. While a satisfactory and reliable dimensioning has been achieved via static simulations (first by Ferrari (2010), then refined by De Simoi (2011)), these are not suitable for testing the effectiveness of the controller. First, the control system acts on the dynamics of the vehicle, and a static simulation is not able to faithfully reproduce its behaviour; secondly, this kind of application has a critical impact on the final user, hence a model of the rider has to be considered to avoid safety issues whit the human driver. Hence it is clear the importance of implementing a *virtual environment* where dynamical models of both the vehicle and

the rider are included, so that reliable feedbacks can be obtained and analyzed to tune at best the algorithms before the "real-world" applications.

In the next sections, the virtual environment implemented for testing the control strategies will be presented (Beghi, Maran, and De Simoi, 2012c,d), highlighting some of the crucial features and motivating the implementative choices. Simulation results with both the proposed control algorithms will be analyzed and discussed. Finally, we will report the results of an on-track test with the real prototype, held on the PIAGGIO track in Pontedera (Italy). These results will regard only the heuristic control solution, that has been easier and faster to implement within the due date of the project.

4.1 Virtual environment

The implemented virtual environment has two main goals:

1. provide a flexible, easily integrable tool for *dynamic simulation* of a (simplified) model of the hybrid motorbike;

2. test the proposed control algorithms on a realistic scenario.

The core of the simulation tool is VI-BIKEREALTIME, a professional modeling, post processing and real-time analysis environment for motorcycle models. One of its main features is the SIMULINK interface, which allows to perform software-in-the-loop and hardware-in-the-loop activities: together with the availability of a great number of customizable parameters and output channels (more than 350 outputs), the tool is able to provide all the quantities needed to evaluate the dynamic behaviour of the vehicle.

The key feature of the SIMULINK interface is the presence of a *virtual rider*, which emulates the behaviour of a real driver and allows to close the loop of the dynamic simulation, giving a reliable feedback on the performance of the control strategies in a realistic scenario, before the "real-world" tests. In this sense, it is important for our purposes to *limit* the performance of the virtual driver, implemented as a *maximum performance* driver, to emulate situations of city-driving, as to build ad-hoc tracks to replicate realistic paths; a SIMULINK model of the electrical machine has been implemented to get a complete dynamic system representing the hybrid vehicle, with particular efforts on the representation of the power management system. In figure 4.1 the complete SIMULINK scheme is reported.

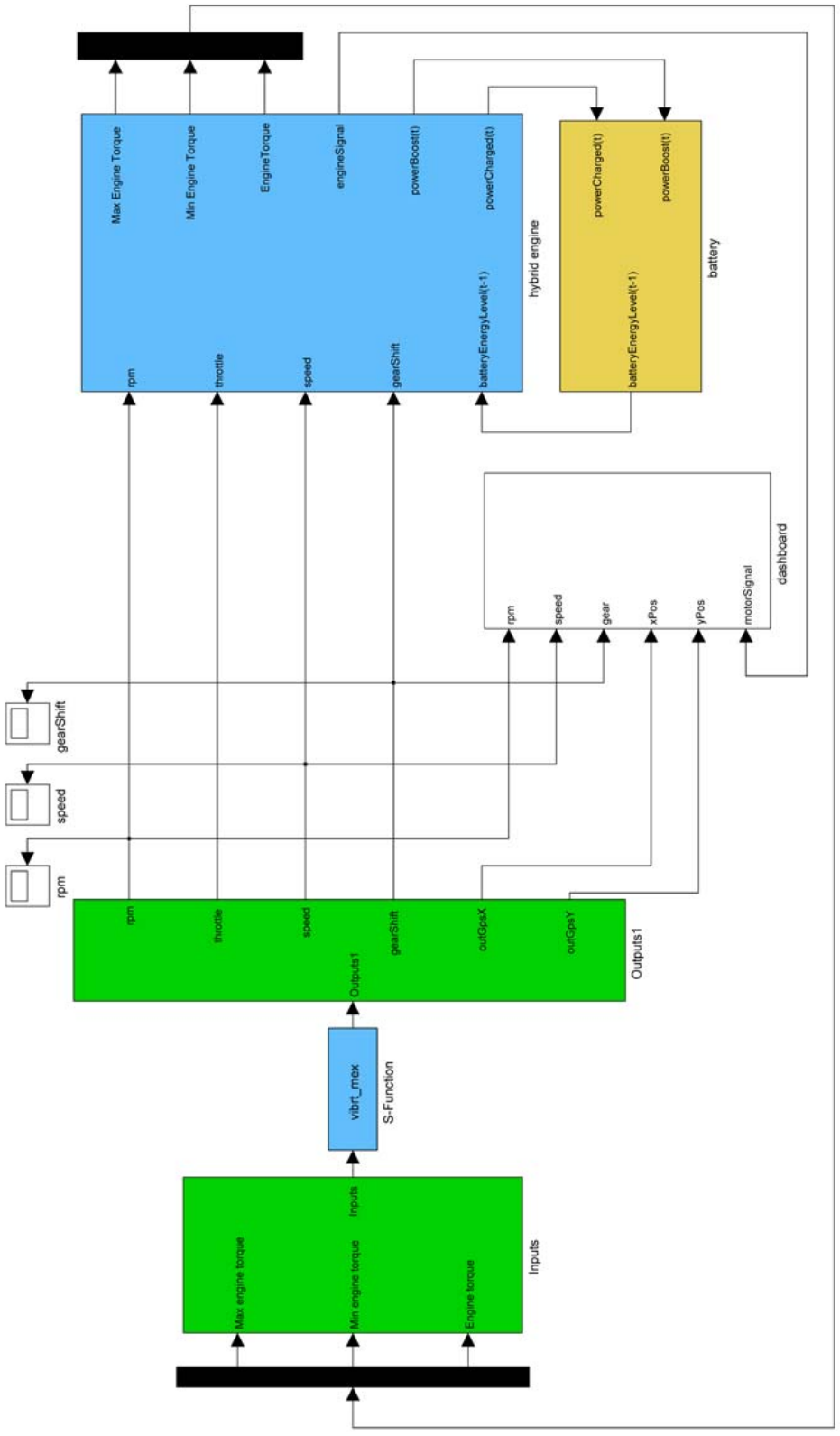


Figure 4.1: SIMULINK scheme of the virtual environment. The green/blue blocks on the left identify the motorbike and rider models; the yellow and blue blocks on the right are the batteries and hybrid engine model, respectively.

4.1.1 Virtual rider

The virtual rider is able to "ride" a virtual vehicle model through a number of different manoeuvres in order to allow users to accurately evaluate the dynamic behaviour. The main advantage is the capability of dealing with closed-loop manoeuvres, which allows to test the effectiveness of control algorithms from the point of view of the final user (the rider): this is achieved by a combination of a static and an accurate *dynamic solver* (Frezza et al., 2003) that, given the model parameters of the motorbike and the track, is able to solve in real-time a set of equations which emulate a real-driver behaviour (Frezza and Beghi, 2006, 2003; Frezza, Beghi, and Saccon, 2004; Saccon, Hauser, and Beghi, 2008), in terms of applied longitudinal and lateral forces, driveline following and management of crucial elements as steering, throttle, brake and many more.

The virtual rider must be provided with *speed* and *gear-shifting* references for a specific track: since it is implemented to give the maximum performance during the ride, the two references have to be manipulated in order to limit the driver capabilities and emulate a city-ride route.

4.1.2 Tracks

A fundamental aspect of a reliable vehicle simulation is the choice of the track: the driver's and vehicle behaviour are deeply affected by the features of the route. For the target of interest, the track has to be able to catch the main features of urban and extra-urban mobility, typical for this kind application. The provided tool has the capability of building a reliable road model, made up of single small blocks that can be characterized in terms of orientation, regularity, grip and other parameters, and when combined, the principal features of the track are specified, e.g. curves position and inclination angles, overpasses, chicanes and many more. In the design stage, a weighted combination of curvature and tension (local measure of the total path length) determines the trajectory that the virtual driver will follow along the circuit: this is clearly another crucial parameter to specify the virtual rider's behaviour.

4.1.3 Hybrid engine

The hybrid engine, with both the endothermic engine and electric motor, is implemented as an *external* SIMULINK system with respect to the core block representing the motorbike/rider dynamics (figure 4.2). In fact, simply adding the torque supplied by the electric motor to the one given by the *integrated* model of the internal-combustion engine leads to a wrong calculation of the throttle signal and an increasing error in tracking the speed

reference: as stated before, the speed reference must be given in advance to the solver, which at each time step calculates the throttle signal w.r.t. the torque that has to be supplied in the next step to match the desired speed (feed-forward action). From the solver point of view, the *whole* input torque is supplied by the endothermic engine, hence leading to a wrong computation of the throttle signal that reflects on a bad tracking of the speed reference that the internal PID controller is not able to prevent and on wrong behaviour of the virtual rider (e.g. using brake instead of engine braking, or following the wrong trajectory). Implementing the hybrid engine as an external system allows to provide the right torque value to the solver in the feed-forward step, obtaining correct results for throttle and speed signals.

Another remarkable advantage of our implementation is the *throttle partialization* management. Both the engine models are *map-based*, i.e. their dynamics is represented by a collection of look-up tables which relate the main quantities (e.g. RPM to supplied torque). Each engine is provided with maps relating the maximum and minimum suppliable torque to the current RPM value (for the electrical machine, the minimum torque is null, since it doesn't contribute to engine braking): from the throttle value applied on the previous step and the current RPM value, the endothermic torque and the new throttle value are calculated, and the last one is used to scale the electric torque to get the correct value. To compute the partialized torque, a linear interpolation is used, but the implemented model gives the possibility to use more accurate partialization maps, if available.

The subsystem includes a block to select the charge and boost intervals, as described in section 3.2.

4.1.4 Battery pack

The system implemented to model the accumulators follows what has been shown in section 3.1: the polynomial models are integrated and parameterized by specifying the number of cells and their characteristics. The power estimation is implemented too, to integrate this element with the rest of the virtual environment. Note that thanks to the chosen approach it is easy to customize this block to different accumulators model, by updating the correspondent charge and discharge curves.

4.1.5 Graphical User Interface

A GUI has been implemented to set the principal parameters for our test purposes: number of battery cells, track and rider profiles, control strategies, rider's driving style. The GUI

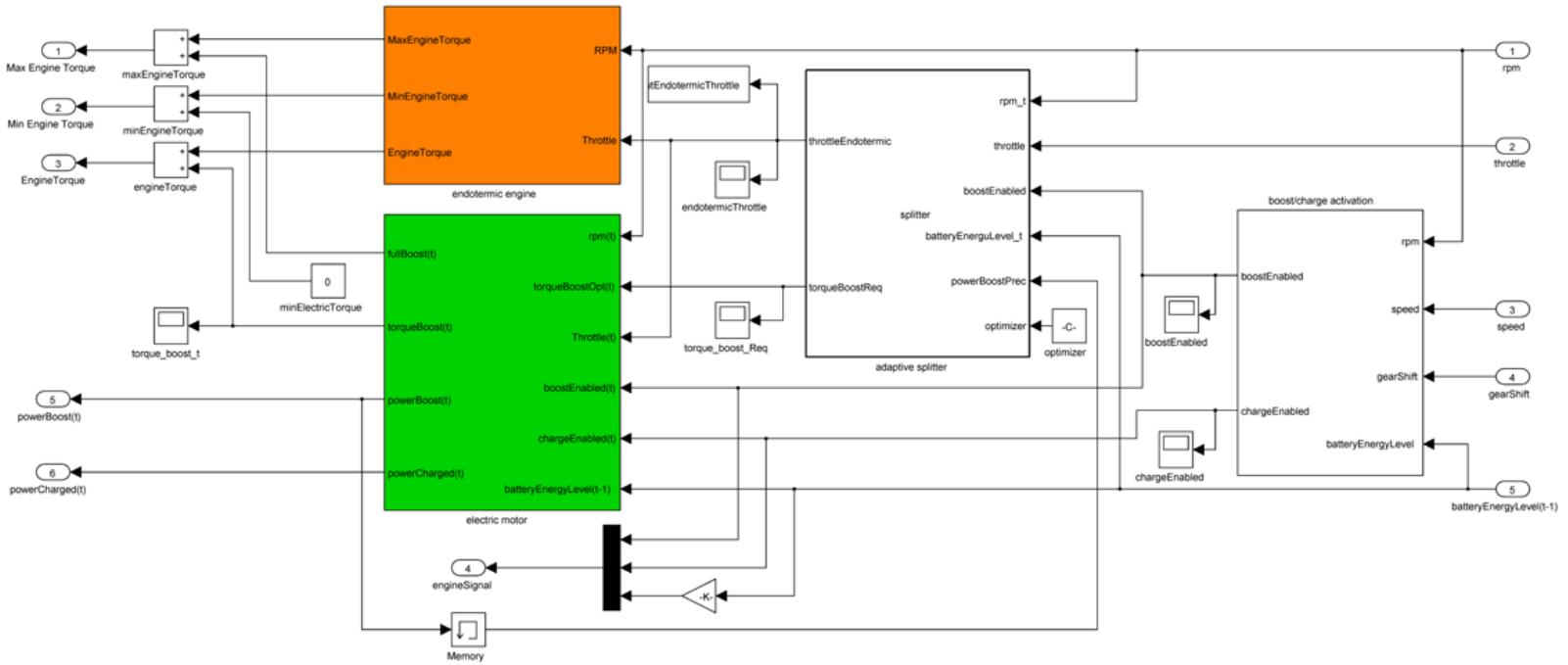


Figure 4.2: Detail of the SIMULINK scheme of the hybrid engine. The orange block is the endothermic engine, the green one the electric motor; on the right, the blocks model the activation strategy and the optimal controller.

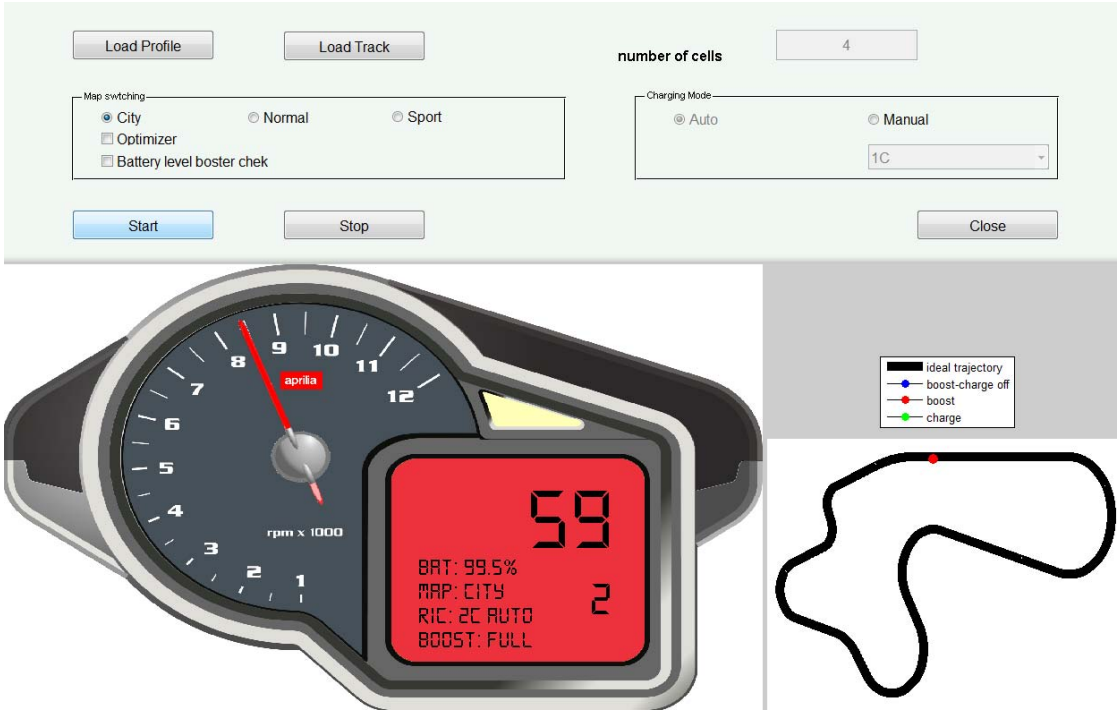


Figure 4.3: Graphical User Interface screenshot.

allows to modify the control strategy and driving behaviour in real-time, while a user friendly representation of the hybrid motorcycle current operation mode and set-up is provided through a dashboard (Figure 4.3).

4.2 Simulation results

The proposed control strategies have been tested on the previously described virtual environment. The goal of the simulation is twofold:

1. to test the management of the chosen battery battery-pack, made up of 4 KOKAM Li-Poly cells, on a realistic track;

2. to test the improvements in driving performance due to the electric boost contribution.

The last point cannot be directly verified: as previously stated, the solver of the virtual rider needs a speed reference to track. Hence, in order for the comparisons to be reliable, the rider has to be provided with the same speed reference, whatever the chosen motorbike (pure endothermic or hybrid) and control strategy are. In this sense, the comparison will

be carried out on the *throttle demand* rather than the applied boost: given the same speed profile, a smaller throttle demand corresponds to an improvement due to the boost action, so the target becomes emissions reduction.

The first simulation sees a gear-shift map realized to limit the virtual rider's performance as to emulate the typical RPM profile of urban driving paths, favouring up-shifts to keep low running speeds. The designed track, 1793 meters long, contains both urban and extra-urban features, with close brakings and accelerations phases together with long straights and fast bends: in figure 4.4 the track is shown. The boost and charge intervals are highlighted in red and green respectively, in order to give an idea of the operation-mode selection strategy with respect to the spatial location of the motorbike on the track. In figure 4.5 the same information are reported on the RPM telemetry, where the action of the strategy is more clearly visible in terms of speed variation. Note the behaviour of the gear-shift control that suspends the boost supply during up-shifts, preventing useless energy consumption. In figure 4.6 the two proposed control strategies

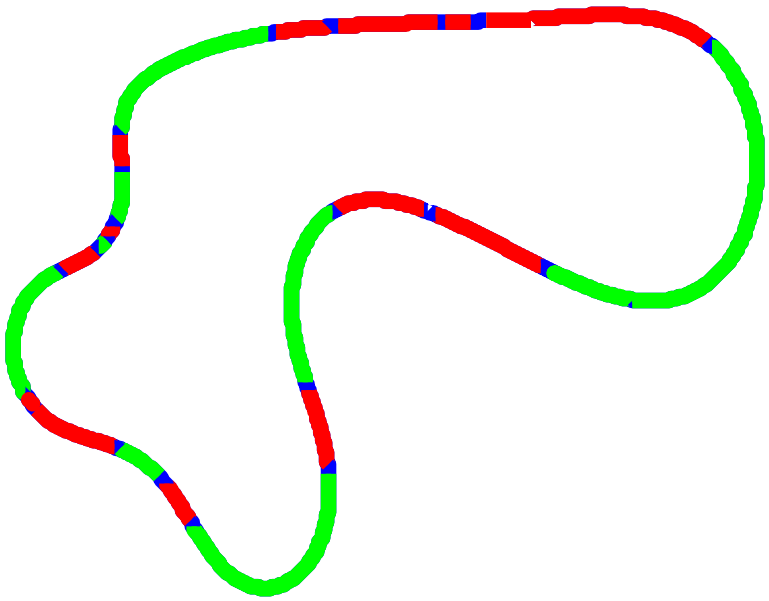


Figure 4.4: Simulation test track: in red the boost intervals, in green the charge ones.

are compared on the same speed reference, together with the behaviour of the pure endothermic motorbike. By observing the throttle request, the demand of the hybrid vehicle is smaller than that of the endothermic one during almost all the accelerations, as can be expected, and this correspond to different applied boost torque values. It is interesting to notice the different behaviour of the two compared control strategies, in term of electrical machine exploitation and consequent contribution to the total torque.

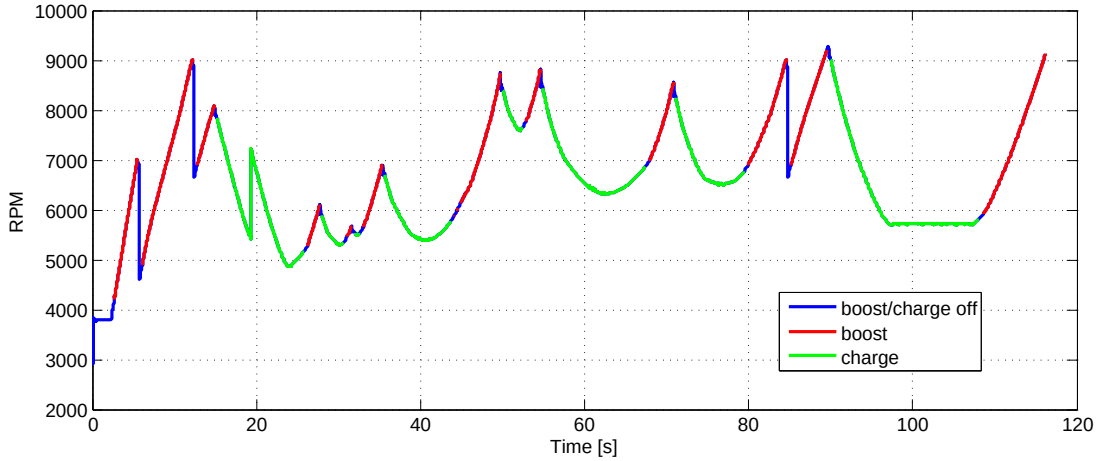


Figure 4.5: RPM telemetry on test track: in red the boost intervals, in green the charge ones.

The advantages of the adaptive torque splitter are evident: the electrical machine is much more exploited, with remarkable throttle demand, and consequently emission, reduction, in particular during long accelerations. The exploitation of the electric motor in terms of boost supply is also more regular using the optimal control strategy, thanks to the effectiveness of the cost function. For what concerns the effect on the accumulators management, the State Of Charge of the battery in figure 4.7b shows the battery usage with the two different strategies. At the start of the simulation the batteries are full-charged; the optimal control has an imposed SOC_{ref} value of 0.95. It is evident how the two different strategies allow for different usage of the batteries, as expected due to the different torque supply behaviour. The reason can be found in the different control philosophy: adaptive boost only acts on the electric motor independently of operation of the endothermic engine. In particular, the latter operates at its maximum capabilities before the intervention of the electric one, that supplies the remaining torque contribution to meet the overall hybrid torque profile. On the other hand the optimal strategy also considers the throttle command, so it is able to split the contribution so that to exploit at best the electric motor, with consequent fuel and emission savings. All this is carried on while keeping the SOC at satisfactory and safe values, even if it is less conservative than the adaptive boost case. In both cases the low exploitation is evident, with a SOC kept always over 95%. This is due to the fact that the electric engine is designed to provide a strong contribution at low RPM ($1000 \div 4000$) values as reported in the hybrid torque profile (figure 2.8), while the virtual driver is conceived for racing purposes, and even with a gear-map that favors lower rotational speeds it is hard to keep the vehicle at such low RPM. To this aim it is interesting to observe the initial phases of the telemetry, where

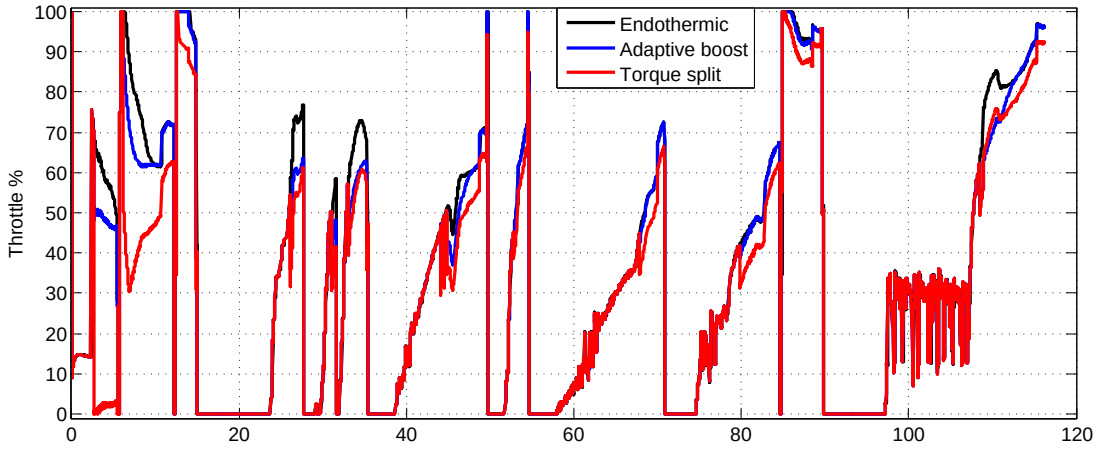
the lap starts and RPM < 5000 (figure 4.7): in this phases the contribution of the electric machine is more effective, and it is clear how the optimal strategy is able to manage the batteries and the throttle partialization to obtain the best performance. Hence, the SOC remains high because the conditions where the electric machine acts as a generator are much more likely to be verified, so if the charge dynamics is slower, the stored energy results to be much higher than the elapsed one.

To understand the different battery exploitation, a test on the same track has been carried out by performing 10 consecutive laps, with a starting SOC equal to 80% and an initial $SOC_{\text{ref}} = 72\%$: results are in figure 4.8. The adaptive boost has an almost constant trend (with a sudden increase at about 420 s that is probably due to imperfection on the linking of different speed references for the virtual vehicle), and this highlights how this strategy is more suitable for slower driving style. The optimal split is able to exploits the battery in these conditions too, and the state of charge is in any case kept at a safer level with respect to the imposed limitation. Note that the telemetry does not make necessary the SOC_{ref} value to be adapted.

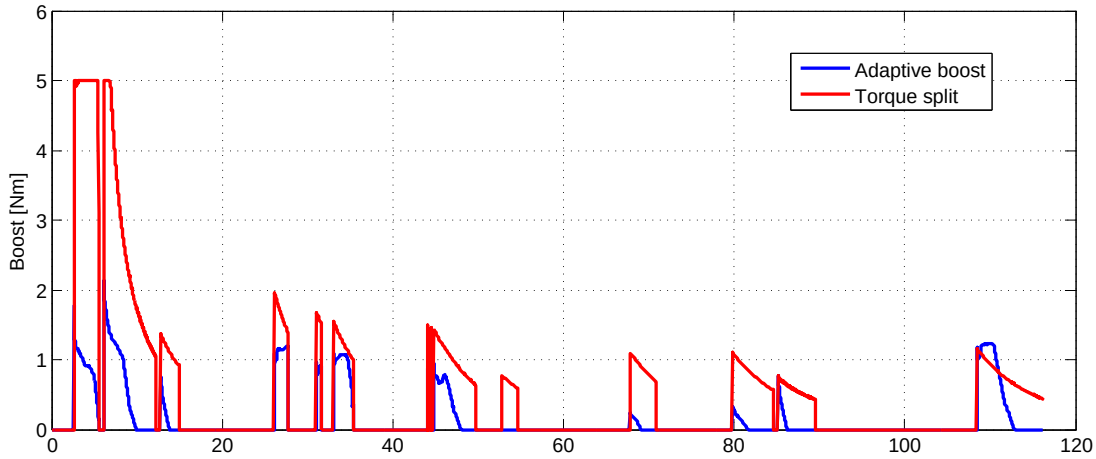
Finally, a test has been performed on a digital version of a real racing circuit, namely the Silverstone (UK) track (figure 4.9). In this case the rider has been let free to behave with maximum performance, to analyze a case when the hybrid vehicle is used in a racing context. The RPM trend on two laps is depicted in figure 4.10, and its large values (always in the range $6000 \div 12000$) will probably make hard to exploit the electric motor. The results in terms of throttle, torque and SOC are reported in figure 4.11. As expected, in this case the difference in throttle request between the two strategies is reduced, since such high rotational speeds are outside the operational range of the electric motor. In any case, the comparison of torque contribution and residual state of charge highlights the fact that the optimal control strategy keeps exploiting the electrical machine as much as possible, with respect of the batteries constraints. In particular, also in this case it is interesting to analyze the behaviour of the SOC dynamics: while the splitter tries to exploit the electrical machine even in this hard conditions, using the stored energy, the adaptive boost strategy is only able to charge the accumulators (an initial SOC value equal to 95% has been set to examinate this behaviour). This confirm the fact that this controller, even if simpler to implement, performs in a more conservative way in terms of electrical machine exploitation.

Remark 4.2.1. Note that all the simulations have been carried out in real-time, or even faster, thus demonstrating the applicability of these power management system strategies on real vehicles, with ECU-optimized implementation.

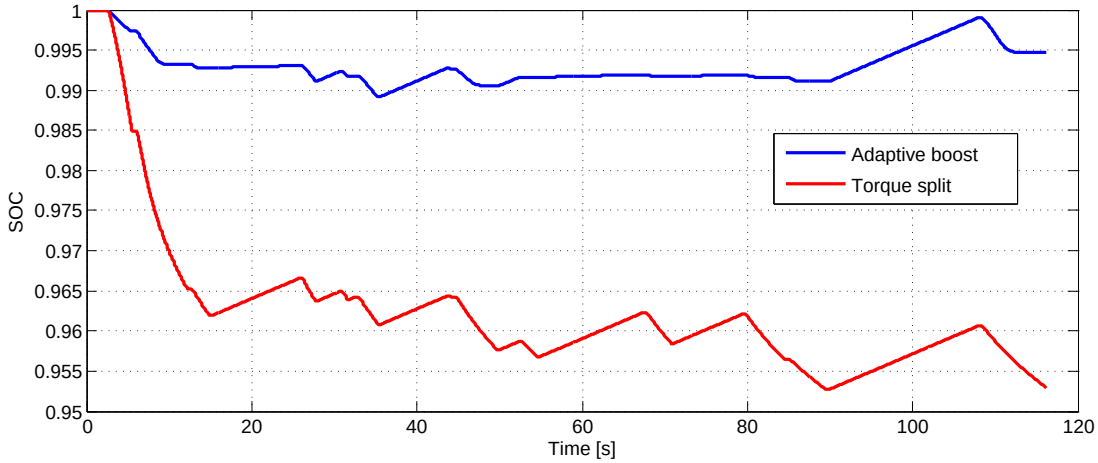
These tests have proved that the proposed control strategies are reliable, even if they can have a conservative behaviour. In particular, they grant the stability of the vehicle even with a rider in the loop, hence they can be safely tested in practice.



(a) Throttle demand.

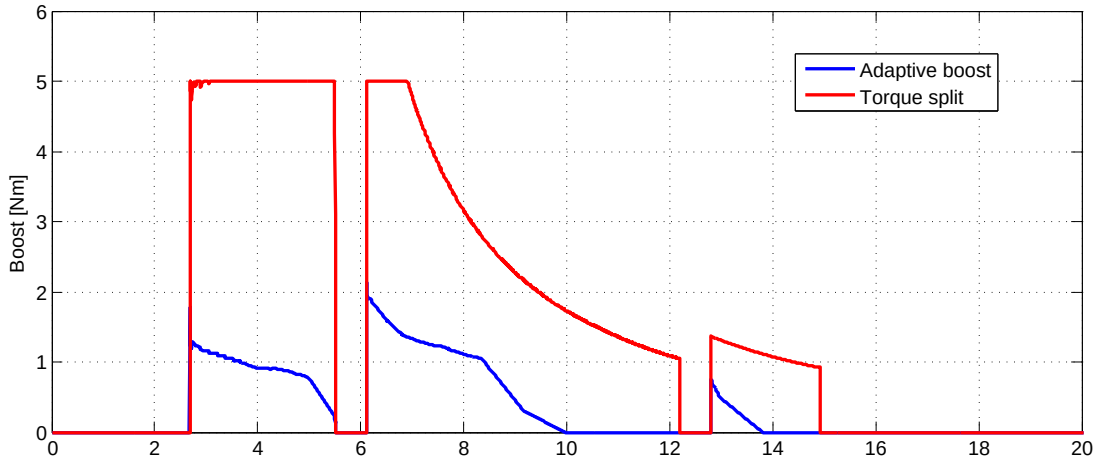


(b) Boost provided.

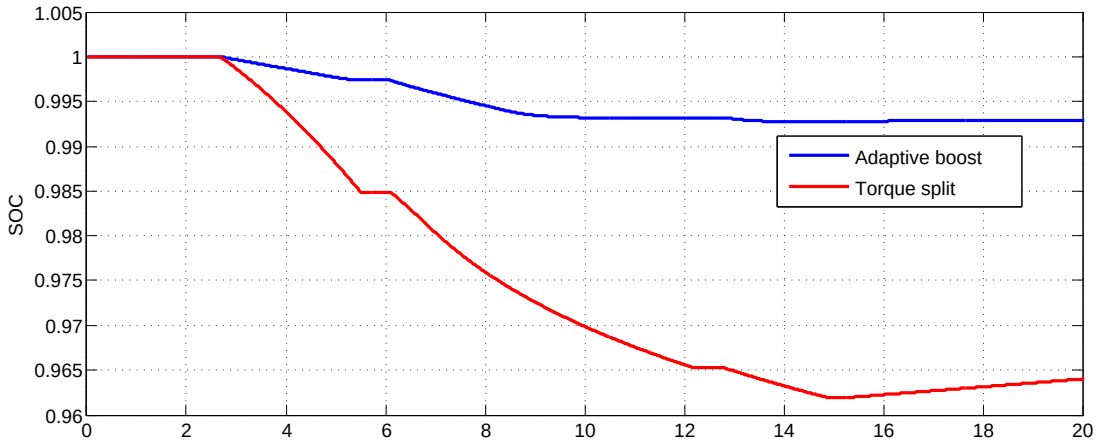


(c) Trend of SOC.

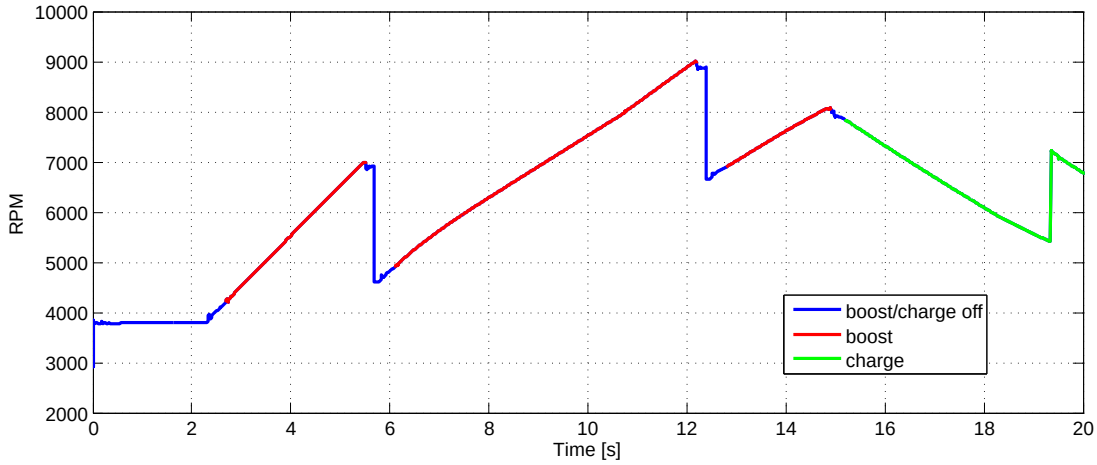
Figure 4.6: Comparison of throttle, boost and SOC telemetries on one lap on the test track: the torque split strategies shows better exploitation of the electric machine.



(a) Boost provided.



(b) Trend of SOC.



(c) Trend of RPM.

Figure 4.7: Comparison of boost, SOC and RPM telemetries on one lap on the test track: the detail on the first part of the telemetry, with a wider range of RPM, gives and advice of the less conservative exploitation of the electric machine provided by the optimal strategy.

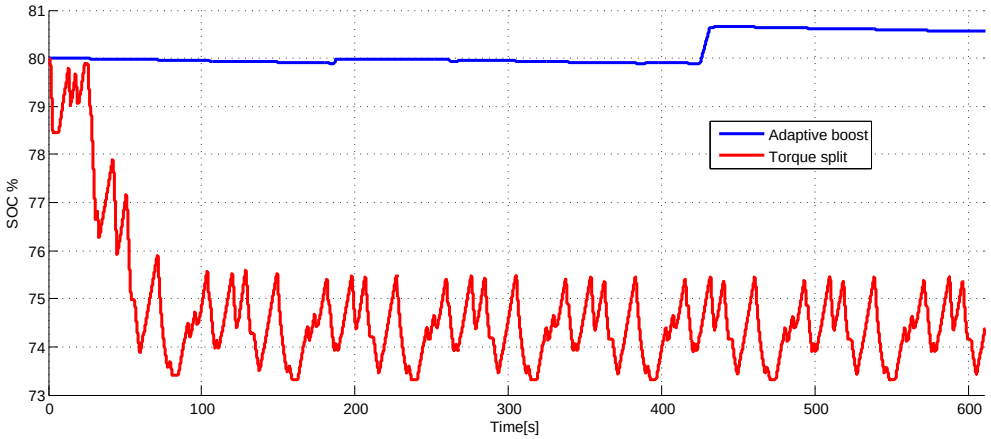


Figure 4.8: *SOC* trend on 10 laps performed on the test track: the splitter strategy exploits the electrical machine more efficiently.

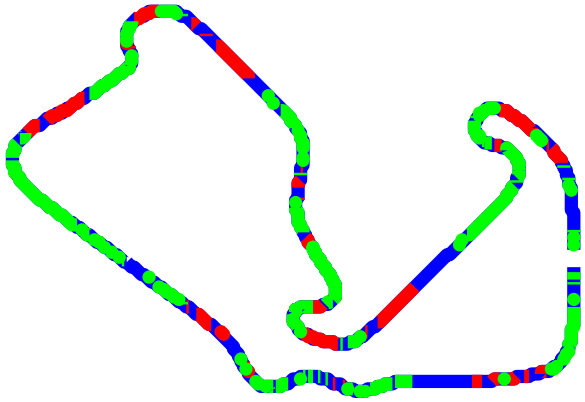


Figure 4.9: Silverstone racing circuit: in red the boost intervals, in green the charge ones.

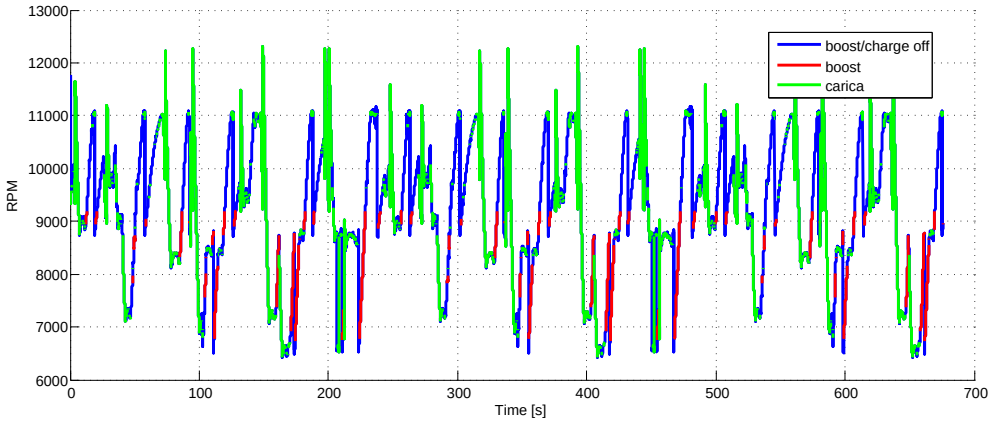
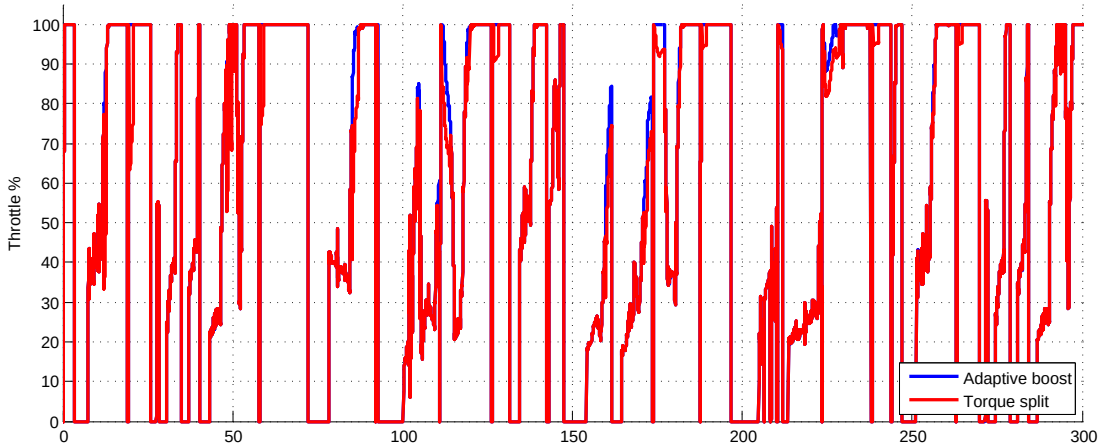
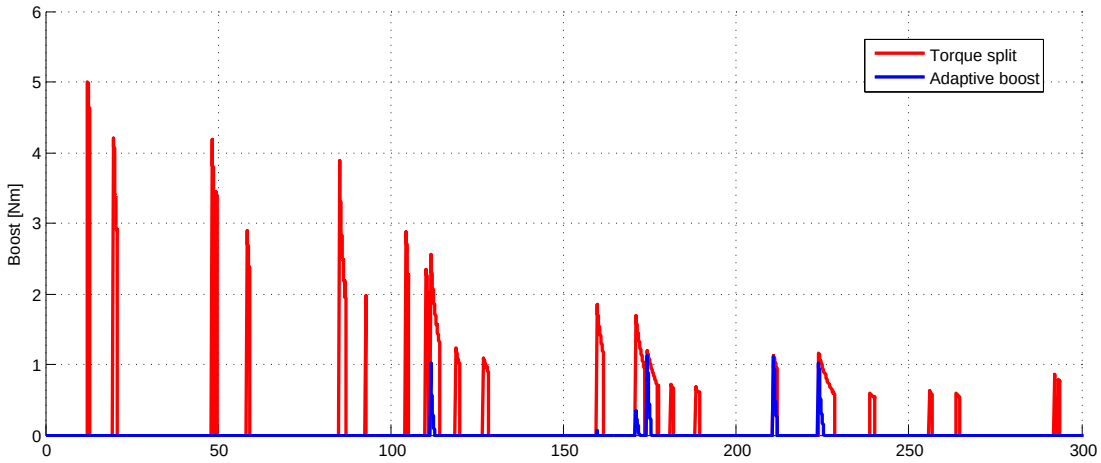


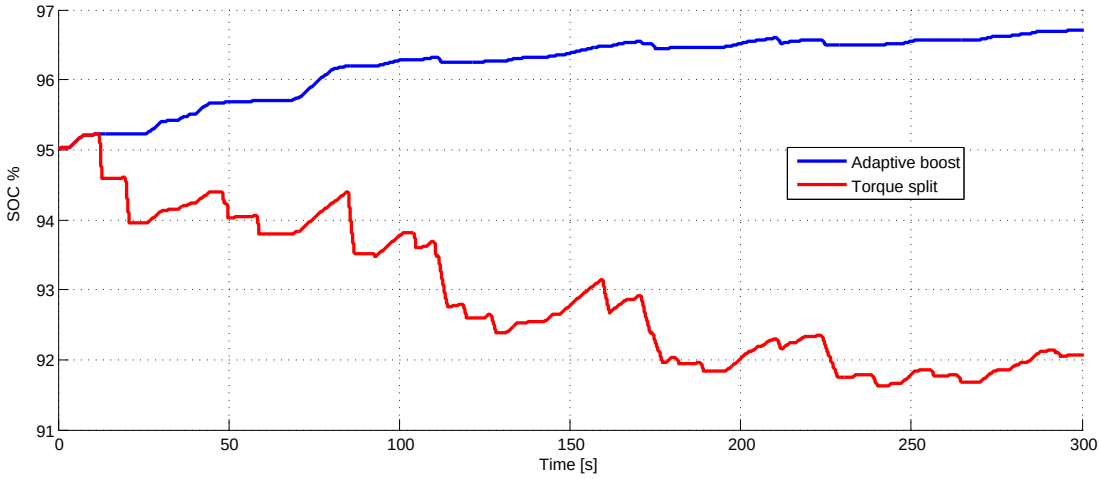
Figure 4.10: rpm telemetry on 2 laps performed on Silverstone track in racing conditions: high RPM values limit the usage of the electric motor.



(a) Throttle demand.



(b) Boost provided.



(c) Trend of SOC.

Figure 4.11: Comparison of throttle, boost and SOC telemetries on two laps on Silverstone track: the racing conditions result in a lower usage of the electric machine. The behaviour of the split strategy helps its exploitation while adaptive boost is almost inactive.

4.3 On-track results

After the validation with the previously depicted simulations, the first of the two control strategies has been implemented in the ECU of the real prototype, thanks to its relative simplicity that makes it more suitable for a fast realisation and application.

The prototype has been realized by integrating the designed electric machine in place of the alternator, that has been removed: the electric engine will fulfill its tasks. The inverter has been installed on the back of the motorbike, inside a box. This is acceptable for prototyping purposes, but it will have to be relocated (and redesigned) in the prospective engineering. The accumulators pack has been located under the seat, together with the electrical filters and the ECU. Figure 4.12 shows a picture of the modified RS4 125.

Figure 4.12: Hybrid RS4 125 prototype.

In the first instance, the behaviour of the hybrid engine has been characterized on the test bench, hence the results are briefly reported. Figure 4.13 shows the results of the measurements of supplied torque and power. The *inertial* bench test measures the acceleration imparted to the mass of the roller, without activating the brake. This kind of test is very fast, but does not guarantee the correctness of the measured profile, since the inertia of the roller hides the uncertainties. However, given that the objective of the measure was not to characterize the real torque curve of the motor but only to evaluate the contribution of the electric motor, this uncertainty can be neglected. The comparison of the performance of the hybrid engine with the endothermic one confirms how the most significative contribution is at low RPM values and decreases while the rotational speed

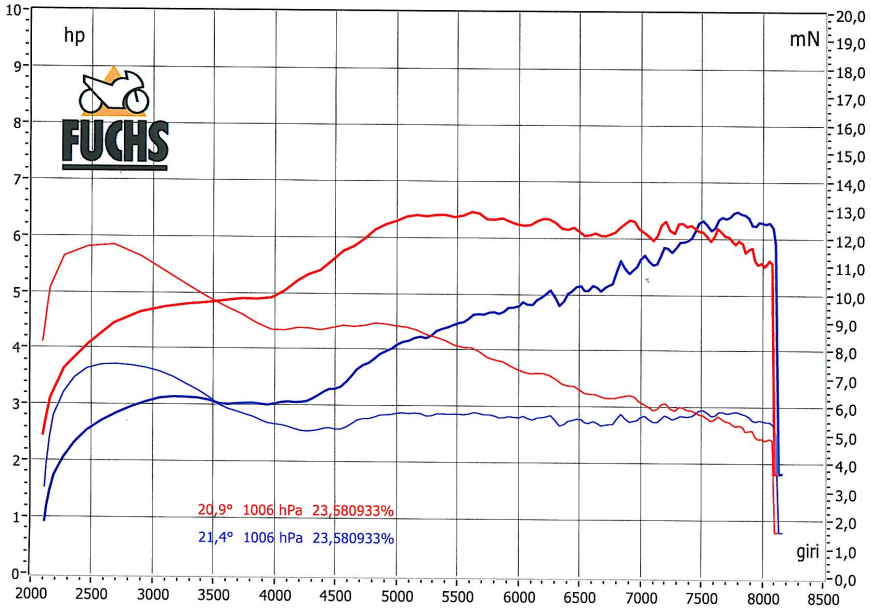


Figure 4.13: Inertial test bench characterization: in red, the hybrid torque and power profiles, in blue the endothermic ones.

increases. The power curve is of particular interest, since it shows how the peak value is reached almost 2500 RPM before the case with the endothermic engine alone, providing a noticeable performance improvement.

The charge phase has been tested too, by verifying that during charging phase the voltage at the battery pack terminals increases: this was confirmed by measurements with a voltmeter.

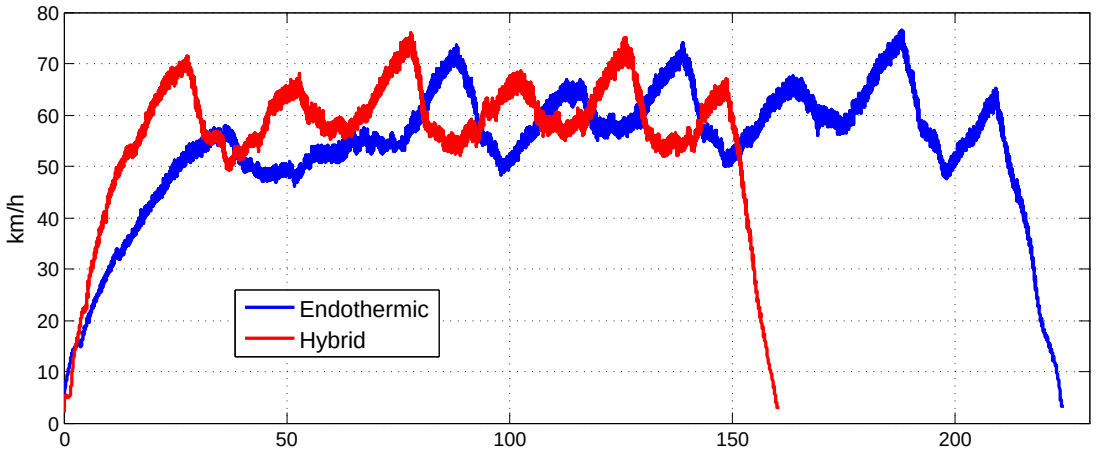
4.3.1 PIAGGIO test track

The prototype has finally been tested on a the test track in Pontedera: a sketch can be seen in figure 4.14. Three different setups were used: endothermic only, charge only and hybrid. For what concerns the batteries, there has not been the possibility of directly measuring the behaviour of the voltage of the accumulators, so it is not possible to evaluate the performance of the battery management and the SOC variations. In any case the control signal applied to the electric motor by the inverter is available, hence it is possible to distinguish when the different phases took place.

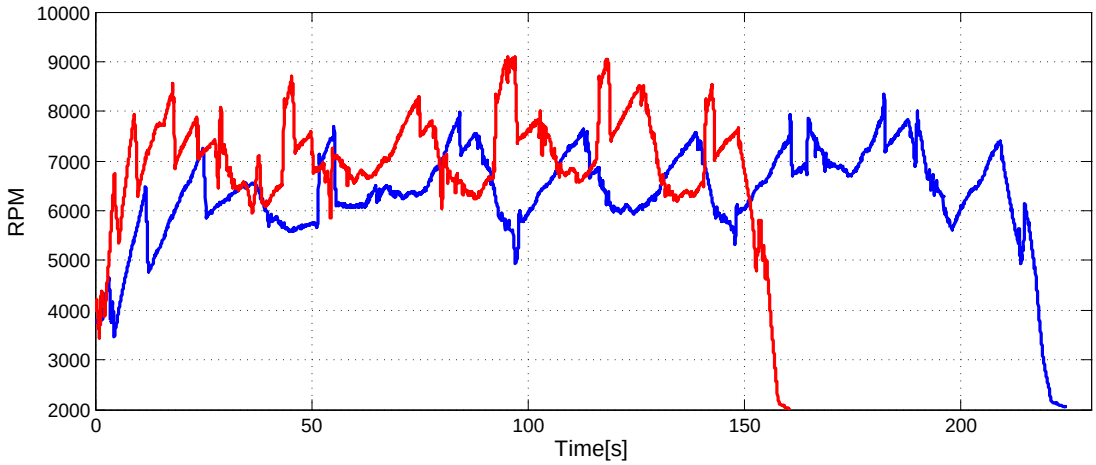
On the other hand, it is interesting to compare the speed performance of the endothermic setup with respect to the hybrid one: in this case the difference can be more easily interpreted than in the virtual environment framework, where the need for fixed speed reference forced to look at the throttle signal. To the contrary, the analysis of throttle

Figure 4.14: PIAGGIO test track.

opening is not significant, given that the vehicle achieved different speed profiles. The comparison is reported in figure 4.15. The improvement provided by the hybridisation is evident: the peak speeds are slightly higher when the electric contribution is active, but the main result is the great increase in acceleration performance, that highlights the effect of boost. In particular, the first part of the telemetries shows the most noticeable difference in terms of acceleration, and this is the consequence of the boost contribution at low RPM value: the depicted trend highlights in fact how only at the start of the test the rotational speed is low enough (less than 5000 RPM) to take full advantage of the electric motor contribution, and the correspondent first acceleration shows its benefits. The overall lap time is sensibly reduced: 160.5 s versus 224.2 of the endothermic vehicle. Hence, the goal of increasing the ride performance has to be considered achieved.



(a) Speed telemetry.



(b) RPM telemetry.

Figure 4.15: Endothermic and hybrid systems: on-track performance comparison. Note the great contribution in acceleration in the first part of the signal, thanks to RPM values that allow for exploitation of the electric motor.

5
Conclusions

In the second part of this dissertation, the problem of designing and validating a controller for a hybrid sport motorbike has been discussed. Different parts of the hybrid system have been modelized, in particular the accumulators. Their physical characteristics have not been directly measured, due to the lack of a test bench, so a modelization based on the information from datasheet has been proposed. Through this, an estimation strategy for the provided and absorbed power has been implemented.

Two control algorithms have been proposed: the first is an heuristic based on the direct control of the sole electric machine. This strategy sees the intervention of the electric motor only if the endothermic is not able to provide the full torque needed by itself. In this sense it is a conservative controller, from the point of view of electric machine and batteries exploitation. The second one is an optimal control strategy based on the idea of *torque split*: the algorithm acts on the endothermic engine too with the aim of enhancing the use of the electric machine while assuring that the accumulators are not too much

stressed. Both the controllers have adaptive features that help the management with different path profiles.

A virtual, dynamical environment for the test of the hybrid motorcycle has been implemented. The battery model and control strategies have been included in a flexible SIMULINK interface and integrated with a virtual rider tool for testing purpose: simulations have been carried out to validate the battery sizing and the effectiveness of the control, both on stability and power management, with satisfactory results in particular with the optimal control strategy.

Finally, the first of the two control algorithm, easier to realize, has been implemented on the real prototype and tested on a track, with clear improvement in the dynamical performance of the vehicle, in particular at low RPM values.

Next steps will be the improvement of electrical machine and battery models to help improving the reliability of the virtual environment, in order to test different vehicles and different tracks an driving styles. The development of different boost mappings is already on the way, in order to check the performance obtained from the vehicle with more or less conservative exploitation of the electrical powertrain. This feature is already implemented on the virtual environment. Furthermore, studies will be carried on to realize algorithm for the identification of the driving conditions, e.g. using *machine learning* techniques, with the goal of adapting the boost supply to the features of the route and the driving style by automatically selecting the most suitable torque map. Finally, the approach of torque split, which has shown promising results will be refined by introducing more refined system model and cost functions. Model Predictive Control will be discussed as well to manage the electric motor, as proposed by Bolognani, Bolognani, Peretti, and Zigliotto (2009).

Final Remarks

In this dissertation, two innovative control applications in the automotive field have been considered.

The MPC-based Motion Cueing Algorithm was inspired by the need of more effective procedures to exploit dynamic simulator platforms, with respect to more classical approaches. After a review of the existing literature about similar ideas, a thorough analysis of the existing studies on vestibular system has been carried out, with the aim of deriving reliable models to be integrated in the MPC framework (Beghi, Bruschetta, Maran, Minen, Baseggio, and Pozzi, 2011). The complete system for the algorithm has seen the coupling with a simple mechanical model, in order to be as general as possible, easy to be integrated in existing devices by generating *references* for the motion controller (usually PLCs), rather than control the actuators themselves (Baseggio, Beghi, Bruschetta, Maran, and Minen, 2011). If further information on these components are available, the model can be revised and customised. The problem has been formulated as a Quadratic Programming problem, for which many solvers are available. The chosen one is based on *Active Set* strategy, and is able to work in real-time. The overall procedure introduces a new concept for the tuning phase, where constraints and weights of a cost function are manipulated to get the desired behaviour. This methodology has the advantage of being intuitive and fast to adapt to the needs of the specific driver (Beghi, Bruschetta, and Maran, 2012a). Finally, the proposed MC algorithm has been implemented and tested on a real platform, with satisfactory results (Beghi, Bruschetta, Maran, Minen, Baseggio, and Pozzi, 2012b).

The control strategies for the hybrid motorbike have been elaborated with the goal of obtain an increase in terms of low-RPM torque performance, avoiding safety issues and excessive battery consumption. To this aim, after a study of the specific components of the prototype, a model for the accumulators has been derived, that makes use of the

information from the data-sheets. Two controllers have been proposed, a simpler, heuristic one that uses the electrical machine when the endothermic engine can not provide the requested torque, and a more sophisticated one based on a optimal *torque split* control strategy, that manages both the powertrains to achieve the best combination possible in terms of performance and fuel savings, while preserving the battery pack from excessive stress (Beghi et al., 2012c). A simulation environment has been implemented to test the proposed strategies: it integrates the battery model and a virtual rider that closes the loop in order to validate the algorithms (Beghi et al., 2012d). Finally, the heuristic strategy has been implemented in the Electronics Control Unit of the prototype and tested on track, with satisfactory results.

Both the topics have studied from scratch, innovative algorithms have been designed and after being validated via simulation, implemented and tested on the field.

References

ACADO Toolkit . Nonlinear optimisation solver, 2012. URL www.acadotoolkit.org.

Ansible Motion . Motion simulators manufacturer, 2012. URL ansiblemotion.com.

Aprilia S.p.A. . Official web-site, 2012. URL www.aprilia.com.

Augusto B. and Loureiro . Motion cueing in the chalmers driving simulator: A model predictive control approach. Master's thesis, Chalmers University of technology, 2009.

Bartlett R. A. and Biegler L. T. Qpschur: A dual, active-set, schur-complement method for large-scale and structured convex quadratic programming. *Optimization and Engineering*, 7:5–32, 2006. ISSN 1389-4420.

Bartlett R. A., Biegler L. T., Backstrom J., and Gopal V. Quadratic programming algorithms for large-scale model predictive control. *Journal of Process Control*, 12(7):775 – 795, 2002.

Baseggio M., Beghi A., Bruschetta M., Maran F., and Minen D. An mpc approach to the design of motion cueing algorithms for driving simulators. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 692 –697, oct. 2011.

Beghi A., Bruschetta M., and Maran F. A real time implementation of mpc based motion cueing strategy for driving simulators. In *Decision and Control (CDC), 2012 51st International IEEE Conference on*, dec. 2012a.

Beghi A., Bruschetta M., Maran F., Minen D., Baseggio M., and Pozzi M. Study on the next generation motion cueing for driving simulator. In *JSAE Annual Congress*, may 2011.

Beghi A., Bruschetta M., Maran F., Minen D., Baseggio M., and Pozzi M. A model-based motion cueing strategy for compact driving simulation platforms. In *Driving Simulation Conference*, sep. 2012b.

Beghi A., Maran F., and De Simoi A. A simulation environment for assessing power management strategies in hybrid motorcycles. In *9th Internation Conference on Modeling Optimization and SIMulation*, june 2012c.

Beghi A., Maran F., and De Simoi A. A virtual environment for the design of power management strategies for hybrid motorcycles. In *3rd Internation Conference on Circuits Systems Control and Signals*, sep. 2012d.

Beghi A., Nardo L., and Stevanato M. Observer-based discrete-time sliding mode throttle control for drive-by-wire operation of a racing motorcycle engine. *Control Systems Technology, IEEE Transactions on*, 14(4):767 – 775, july 2006.

Bemporad A., Morari M., Due V., and Pistikopoulos E. N. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 – 20, 2002.

Bernd F. *Polynomial Based Iteration Methods for Symmetric Linear Systems*. Society for Industrial and Applied Mathematics, 2011.

Bolognani S., Bolognani S., Peretti L., and Zigliotto M. Design and implementation of model predictive control for electrical motor drives. *IEEE Transactions on Industrial Electronics*, 56(6):1925–1936, June 2009.

Boyd S. and Vandenberghe L. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

Camacho E. F. and Bordons C. Nonlinear model predictive control: An introductory review. In **Findeisen R., Allgöwer F., and Biegler L.**, editors, *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, pages 1–16. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72698-2.

Chan C. C. The State of the Art of Electric, Hybrid, and Fuel Cell Vehicles. *Proceedings of the IEEE*, 95(4):704–718, April 2007. ISSN 0018-9219. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4168013.

Chan C., Bouscayrol a., and Chen K. Electric, Hybrid, and Fuel-Cell Vehicles: Architectures and Modeling. *IEEE Transactions on Vehicular Technology*, 59(2):589–598, February 2010. ISSN 0018-9545. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5276874.

Chen M. and Rincon-Mora G. Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance. *IEEE Transactions on Energy Conversion*, 21(2):504–511, June 2006.

Concurrent Real-Time . Embedded & real-time linux, 2012. URL real-time.ccur.com.

Conrad B. and Schmidt S. F. A study of techniques for calculating motion drive signals for flight simulators. Technical report, NASA, July 1971.

Dagdelen M., Reymond G., Kemeny A., Bordier M., and Maīzi N. Model-based predictive motion cueing strategy for vehicle driving simulators. *Control Engineering Practice*, 17(9):995–1003, 2009.

Dagdelen M., Reymond G., Kemeny A., Bordier M., and Maïzi N. MPC Based Motion Cueing Algorithm: Development and Application to the ULTIMATE Driving Simulator. In *Proceedings of the Driving Simulation Conference 2004 Europe, Paris, France*, pages 221–233, 2004.

D'Ambrosio D. Algoritmi di motion cuening per simulatori di veicolo. Master's thesis, University of Padova, Control System Engineering, October 2010.

De Nicolao G., Magni L., and Scattolini R. Stability and robustness of nonlinear receding horizon control. In **AllgÃűwer F. and Zheng A.**, editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 3–22. BirkhÃď user Basel, 2000. ISBN 978-3-7643-6297-3.

De Simoi A. Ambiente virtuale per lo sviluppo di motocili ibridi. Master's thesis, University of Padova, 2011.

Debert M., Colin G., Mensler M., Chamaillard Y., and Guzzella L. Li-ion battery models for hev simulator. *Les Rencontres Scientifiques de l'IFP: Advances in Hybrid Powertrains*, 2008.

Delta Tau Data Systems . Geo brick, 2012. URL www.deltatau.co.uk/geo-brick-lv-ims2.html.

Dougal R. A. Dynamic lithium-ion battery model for system simulation. *IEEE Transactions on Components and Packaging Technologies*, 25(3):495–505, 2002.

Ehsani M., Gao Y., and Miller J. M. Hybrid Electric Vehicles: Architecture and Motor Drives. *Proceedings of the IEEE*, 95(4):719–728, April 2007. ISSN 0018-9219. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4168020.

Emadi a., Rajashekara K., Williamson S., and Lukic S. Topological Overview of Hybrid Electric and Fuel Cell Vehicular Power System Architectures and Configurations. *IEEE Transactions on Vehicular Technology*, 54(3):763–770, May 2005. ISSN 0018-9545. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1433222.

European Union . The european green cars initiative, 2008a. URL www.green-cars-initiative.eu.

European Union . Work programme 2011, 2008b. URL ftp.cordis.europa.eu/pub/fp7/docs/wp/cooperation/transport/g-wp-201101_en.pdf.

European Union . Background on energy in europe. European Council, February 2011.

Fernandez C. and Goldberg J. M. Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. i. response to sinusoidal stimulation and dynamics of peripheral vestibular system. *Journal of Neurophysiology*, 34(4):661–675, July 1971.

Fernandez C. and Goldberg J. M. Physiology of peripheral neurons innervating otolith organs of the squirrel monkey. iii. response dynamics. *Journal of Neurophysiology*, 39 (5):996–1008, September 1976.

Ferrari M. Propulsione Ibrida Motociclistica: studio di fattibilità per la realizzazione di un prototipo. Master's thesis, Padova, 2010.

Ferreau H. *Model Predictive Control Algorithms for Applications with Millisecond Timescales*. PhD thesis, K.U. Leuven, 2011.

Ferreau H., Bock H., and Diehl M. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18 (8):816–830, 2008.

Frezza R. and Beghi A. A virtual motorcycle driver for closed-loop simulation. *Control Systems, IEEE*, 26(5):62–77, 2006.

Frezza R., Beghi A., and Saccon A. Model predictive for path following with motorcycles: application to the development of the pilot model for virtual prototyping. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 767 –772 Vol.1, dec. 2004.

Frezza R., Saccon A., and Bacchet D. Smartdriver: a sensor based model of a car driver for virtual product development. In *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, volume 1, pages 366 – 370, 2003.

Frezza R. and Beghi A. Simulating a motorcycle driver. In **Kang W., Borges C., and Xiao M.**, editors, *New Trends in Nonlinear Dynamics and Control and their Applications*, volume 295 of *Lecture Notes in Control and Information Sciences*, pages 175–186. Springer Berlin / Heidelberg, 2003.

Fu L., Ozguner U., Tulpule P., and Marano V. Real-time energy management and sensitivity study for hybrid electric vehicles. In *American Control Conference (ACC), 2011*, pages 2113–2118. IEEE, 2011.

Gill P. E., Gould N. I. M., Murray W., Saunders M. A., and Wright M. H. A weighted gram-schmidt method for convex quadratic programming. *Mathematical Programming*, 30:176–195, 1984a.

Gill P. E., Murray W., Saunders M. A., and Wright M. H. Procedures for optimization problems with a mixture of bounds and general linear constraints. *ACM Trans. Math. Softw.*, 10(3):282–298, 1984b. ISSN 0098-3500.

Goldfarb D. and Idnani A. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33, 1983.

Hartley E., Jerez J., Suardi A., Maciejowski J., Kerrigan E., and Constantinides G. Predictive control of a boeing 747 aircraft using an FPGA. In *Proceedings of the IFAC NMPC '12 Conference*, Noordwijkerhout, NL, August 2012.

Hosman J. C. and Van der Vaart J. C. Vestibular models and thresholds of motion perception. results of tests in a flight simulator. Technical report, Delft University of Technology, Department of Aerospace Engineering, 1978.

Jerez J. L., Goulart P. J., Richter S., Constantinides G. A., Kerrigan E. C., and Morari M. Embedded predictive control on an fpga using the fast gradient method. Submitted to the 12th European Control Conference, July 2013.

Jerez J., Ling K.-V., Constantinides G., and Kerrigan E. Parallel mpc for real-time fpga-based implementation. In **IFAC** , editor, *Proceedings of IFAC World Congress*, pages 1338–1343, Milan, Italy, September 2011a.

Jerez J., Ling K.-V., Constantinides G., and Kerrigan E. Model predictive control for deeply pipelined field-programmable gate array implementation: algorithms and circuitry. *Control Theory Applications, IET*, 6(8):1029 –1041, 17 2012.

Jerez J. L., Constantinides G. A., and Kerrigan E. C. An fpga implementation of a sparse quadratic programming solver for constrained predictive control. In *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*, FPGA '11, pages 209–218, New York, NY, USA, 2011b. ACM. ISBN 978-1-4503-0554-9.

Kennedy R. S., Hettinger L. J., and Lilienthal M. G. *Simulator Sickness*, chapter 15, pages 317–341. Motion and Space Sickness. CRC Press, Boca Raton, Florida, 1988.

Kennedy R. S. and Fowlkes J. E. Simulator sickness is polygenic and polysymptomatic: Implications for research. *The International Journal of Aviation Psychology*, 2(1):23–38, 1992.

Kessels J. T. B. A., Koot M., de Jager B., van den Bosch P. P. J., Aneke N. E. P. I., and Kok D. B. Energy Management for the Electric Powernet in Vehicles With a Conventional Drivetrain. *IEEE Transactions on Control Systems Technology*, 15(3):494–505, May 2007.

Klee V. and Minty G. J. How good is the simplex algorithm? In **Shisha O.**, editor, *Inequalities*, volume III, pages 159–175. Academic Press, New York, 1972.

Kokam . Official web-site. URL www.kokam.com.

Koot M., Kessels J., DeJager B., Heemels W., VandenBosch P., and Steinbuch M. Energy Management Strategies for Vehicular Electric Power Systems. *IEEE Transactions on Vehicular Technology*, 54(3):771–782, May 2005.

Langson W., Chryssochoos I., Raković S., and Mayne D. Robust model predictive control using tubes. *Automatica*, 40(1):125 – 133, 2004.

Lau M., Yue S., Ling K., and Maciejowski J. A comparison of interior point and active set methods for fpga implementation of model predictive control. In *Proc. European Control Conference*, Budapest, August 2009. European Union Control Association.

Li Y., Anderson R., Song J., Phillips A., and Wang X. A nonlinear adaptive observer approach for state of charge estimation of lithium-ion batteries. In *American Control Conference (ACC), 2011*, pages 370–375, San Francisco, CA, USA, 2011. IEEE.

Ling K., Yue S., and Maciejowski J. A fpga implementation of model predictive control. In *American Control Conference*, pages 1950–1955, June 2006.

Ling K., Yue S., and Maciejowski J. Embedded model predictive control (mpc) using a fpga. In *Proceedings of the 17th IFAC World Congress*, pages 250–255, 2008.

Longo S., Kerrigan E. C., Ling K. V., and Constantinides G. A. A parallel formulation for predictive control with nonuniform hold constraints. *Annual Reviews in Control*, 35 (2):207 – 214, 2011.

Maciejowski J. *Predictive control: with constraints*. Pearson education, 2002.

Marangon G. Battery Management Systems for Li-ion Batteries in Hybrid Electric Vehicles. Master's thesis, University of Padova, 2010.

Mayne D. and Michalska H. Receding horizon control of nonlinear systems. *Automatic Control, IEEE Transactions on*, 35(7):814 –824, jul 1990. ISSN 0018-9286.

Meiry J. L. *The vestibular system and human dynamic space orientation*. PhD thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 1965.

Moura S. J., Fathy H. K., Callaway D. S., and Stein J. L. A Stochastic Optimal Control Approach for Power Management in Plug-In Hybrid Electric Vehicles. *IEEE Transactions on Control Systems Technology*, 19(3):545–555, May 2011.

Musardo C., Rizzoni G., and Staccia B. A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1816–1823. IEEE, 2005.

Nahon M. and Reid L. Simulator motion-drive algorithms: A designer's perspective. *J. Guidance*, 13(2), 1990.

Nocedal J. and Wright S. J. *Numerical Optimization*. Springer, 2000.

Optec . qpOASES. Katholieke Universiteit Leuven, 2012. URL www.kuleuven.be/optec/software/qpOASES.

Ormsby C. C. *Model of human dynamic orientation*. PhD thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 1974.

Pannocchia G., Rawlings J. B., and Wright S. J. Fast, large-scale model predictive control by partial enumeration. *Automatica*, 43(5):852 – 860, 2007.

Parrish R. V., Dieudonne J. E., Bowles R. L., and Martin D. J. J. Coordinated adaptive washout for motion simulators. Technical report, NASA, 1975.

Press W. H., Teukolsky S. A., Vetterling W. T., and Flannery B. P. *Numerical Recipes: The Art of Scientific Computing*, chapter Section 10.2. Golden Section Search in One Dimension. Cambridge University Press, New York, 2007.

Pretto P. *The perception and production of speed during self-motion: evidence for non-optimal compensation mechanisms*. PhD thesis, University of Padova, 2008.

Raman S., Sivashankar N., Milam W., Stuart W., and Nabi S. Design and implementation of hil simulators for powertrain control system software development. In *American Control Conference, 1999. Proceedings of the 1999*, volume 1, pages 709 –713 vol.1, 1999.

Reymond G. and Kemeny A. Motion cueing in the renault driving simulator. *Vehicle System Dynamics*, 34(4):249–259, 2000.

Saccon A., Hauser J., and Beghi A. A virtual rider for motorcycles: An approach based on optimal control and maneuver regulation. In *Communications, Control and Signal Processing, 3rd International Symposium on*, pages 243 –248, march 2008.

Shahzad A. and Goulart P. J. A new hot-start interior-point method for model predictive control. In *Proceedings of the 18th IFAC World Congress Milano (Italy), 2011*, volume 18. IFAC, 2011.

Sivan R., Ish-Shalom J., and Huang J.-K. An optimal control approach to the design of moving flight simulators. *Systems, Man and Cybernetics, IEEE Transactions on*, 12(6): 818–827, nov. 1982.

Slob J. J. State-of-the-art driving simulators, a literature survey. DCT 2008.107, Eindhoven University of Technology, August 2008.

Smith K. A. Electrochemical control of lithium-ion batteries. *IEEE Control Systems Magazine*, 30(April):18–25, 2010.

Smith K. A., Rahn C. D., and Wang C.-Y. Model-based electrochemical estimation and constraint management for pulse operation of lithium ion batteries. *IEEE Transactions on Control Systems Technology*, 18(3):654–663, 2010.

Speltino C., Di Domenico D., Fiengo G., and Stefanopoulou A. Comparison of reduced order lithium-ion battery models for control applications. *Proceedings of the 48h IEEE Conference on Decision and Control CDC held jointly with 2009 28th Chinese Control Conference*, pages 3276–3281, 2009.

Steer R. W. J. *The influence of angular and linear acceleration and thermal stimulation on the human semicircular canal*. PhD thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 1967.

Steinhausen W. Observations of the cupula in the ampullae of the semicircular canals of the labyrinth of a living pike. Technical Report 71N25716, NASA, May 1971.

Stewart D. A platform with six degrees of freedom. In *Proceedings of the Institution of Mechanical Engineers*, volume 180, pages 371–386, June 1965.

Straus S. H. New, improved, comprehensive, and automated driver's license test and vision screening system. Technical report, ESRA Consulting Corporation, Arizona Department of Transportation, 2005.

Sung Chul O. Evaluation of motor characteristics for hybrid electric vehicles using the hardware-in-the-loop concept. *Vehicular Technology, IEEE Transactions on*, 54(3):817 – 824, may 2005. ISSN 0018-9545.

Taehun H., Jihoon R., Kihong P., Jeongho H., Kyu H. L., Kangwon L., Soo-Jin L., and Young-Jun K. Development of hils systems for active brake control systems. In *SICE-ICASE, 2006. International Joint Conference*, pages 4404 –4408, oct. 2006.

Telban R., Wu W., Cardullo F., and Center L. R. *Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms*. National Aeronautics and Space Administration, Langley Research Center, 2000.

Telban R. J., Cardullo F. M., and Houck J. A. Motion cueing algorithm development: Human-centered linear and nonlinear approaches. *NASACR*, 213747(May), 2005.

Van Egmond A. A. J., Groen J. J., and Jongkees L. B. W. The mechanics of the semicircular canal. *The Journal of Physiology*, 110(1–2):1–17, December 1949.

VI-Grade . Official website, 2012. URL www.vi-grade.com.

Vincenzi D. A., Wise J. A., Mouloua M., and Hancock P. A. *Human Factors in Simulation and Training*. CRC Press, 2008.

Wang L. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 1848823304, 9781848823303.

Wang Y. and Boyd S. Fast model predictive control using online optimization. *Control Systems Technology, IEEE Transactions on*, 18(2):267–278, 2010.

Wang Y., Zhang W., Wu S., and Guo Y. Simulators for driving safety study: a literature review. In *Proceedings of the 2nd international conference on Virtual reality*, ICVR'07, pages 584–593, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73334-8.

Wright S. J. *Primal-dual Interior-Point Methods*. SIAM, 1997.

Young L. R. and Meiry J. L. A revised dynamic otolith model. *Aerospace Medicine*, 39 (6):606, 1968.

Young L. R. and Oman C. M. Model for vestibular adaptation to horizontal rotation. *Aerospace Medicine*, 40:1076–1080, 1969.

Yurkovich S., Guezennec Y., and Bornatico R. Model-based calibration for battery characterization in hev applications. *2008 American Control Conference*, pages 318–325, 2008.

Zacharias G. L. Motion cue models for pilot-vehicle analysis. Technical Report AMLR-TR-78-2, USA Department of Defense, Alexandria, VA, May 1978.