Università Politecnica delle Marche
Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum "Ingegneria Informatica, Gestionale e dell' Automazione"

Model Predictive Control for Electrical and Mechatronic Systems

Ph.D. Dissertation of:
Luca Cavanini

Advisor:
Prof. Gianluca Ippoliti

Coadvisor:
Prof. Giuseppe Orlando
Prof. Eduardo F. Camacho

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum "Ingegneria Informatica, Gestionale e dell' Automazione"

Model Predictive Control for Electrical and Mechatronic Systems

Ph.D. Dissertation of:
Luca Cavanini

Advisor:
Prof. Gianluca Ippoliti

Coadvisor:
Prof. Giuseppe Orlando
Prof. Eduardo F. Camacho

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

to my family and Elisa

«When you are very young, it happens that you do not know well who you are and what you want from life. Undoubtedly, however, we all have a mysterious thread that sooner or later will end up making us choose what is already latent in us for nature, and will serve to build our personality.»

Walter Bonatti

alla mia famiglia ed Elisa

«Quando si é molto giovani capita di non sapere bene chi si é e che cosa si vuole dalla vita. Indubbiamente peró noi tutti disponiamo di un misterioso filo conduttore che prima o poi finirá per farci scegliere ció che per indole é giá latente in noi, e servirá a costruire la nostra personalitá.»

Walter Bonatti

Acknowledgments

The making of this dissertation would have been impossible without the support from numerous people I encountered throughout my educational, professional, and personal life.
First of all, I would like to express my deepest appreciation and gratitude to my colleague Gionata Cimini, for his teaching and guidance which become my source of inspiration and motivation. I wish to express my gratitude to Luca Romeo, Francesco Ferracuti, Luigi Colombo and Giovanni Manfredi for their helpful suggestions during my research work, their support, patience and encouragement during my research.
My sincere appreciation also goes to Prof. E.F. Camacho of Universidad de Sevilla who gave me the opportunity to work with his research group.

Ancona, November 2017

Luca Cavanini

Abstract

The research activity presented in this thesis concerns the design and the development of control systems based on the prediction of the behavior of the process to be controlled, in order to obtain optimized control performance. The processes taken into account are electrical power converters and mechatronic systems. The use of model-based algorithms to predict the process evolution has been limited for a long time to systems featured by slow dynamics. This issue is related to the time needed to compute the control law, which requires the real time resolution of a constrained optimization problem, that must be always solved in a prescribed amount of time, usually corresponding to the sampling period or even shorter in multi-purpose control units [1]. In recent years the computational the increasing power of control units has allowed the application of predictive control algorithms to complex and faster dynamic processes.

Considering the potential of the predictive control to solve multivariable control problems, thanks to the optimized performance and the effective handling of system's constraints, in this thesis different control algorithms have been developed to solve problems linked to several case studies. Linear Time Invariant (LTI) model-based MPC approaches have been considered for micro-positioning piezoelectric systems and power converters. In this context theoretical conditions to effectively apply MPC for pre-compensed systems on low computational hardware have been given. Control approaches for avionic and marine systems have been also considered. In this context Linear Parameter Varying (LPV) models have been developed and an innovative solution has been studied to reduce the computational effort of LPV-based MPC on embedded control platform. All the above controllers have been tested on real processes, wherever possible, and through highly realistic simulators.

Sommario

L'attività di ricerca presentata in questa tesi riguarda il progetto e lo sviluppo di sistemi di controllo basati sulla predizione del comportamento del processo da controllare, in modo da ottenere prestazioni di controllo ottimizzate. I processi presi in considerazione sono convertitori elettrici di potenza e sistemi meccatronici. L'uso di algoritmi di controllo basati sul modello per predire l'evoluzione del processo è stata limitato a lungo a sistemi caratterizzati da dinamiche lente. Ciò è dovuto al tempo necessario a calcolare la legge di controllo, che prevede la risoluzione di un problema di minimizzazione vincolata da effettuarsi nell'intervallo di campionamento indicato nelle specifiche di progetto (in tempi minori in caso di unità di controllo che gestiscono più processi). Negli ultimi anni, la potenza di calcolo delle schede di controllo embedded è significativamente aumentata, permettendo di applicare le tecniche di controllo predittivo a sistemi caratterizzati da dinamiche veloci.

Considerando le potenzialità del controllo predittivo, in questa tesi differenti leggi di controllo sono state sviluppate per far fronte a problematiche di controllo relative a differenti casi di studio. Approcci MPC basati su modelli lineari tempo invariati (LTI) sono state considerati per il controllo di sistemi di micro-posizionamento piezoelettrici e di convertitori di potenza. In questo ambito sono state presentate le basi teoriche che permettono di applicare con successo tali tecniche su schede di controllo caratterizzate da ridotta potenza computazionale. Controllori per sistemi avionici e marini sono stati inoltre considerati. In questo ambito, sono stati sviluppati modelli lineari a parametri tempo varianti (LPV) rispetto ai quali sono stati sviluppati i sistemi di controllo. In questo ambito è stata sviluppata inoltre un'innovativa tecnica che permette di ridurre il carico computazionale richiesto dagli algoritmi MPC basati su modelli LPV, di modo da permetterne l'implementazione su schede embedded. Tutti i sistemi di controllo sviluppati sono stati testati su processi reali laddove possibile, ed in alternativa su simulatori altamente realistici.

Acronyms

MPC Model Predictive Control

LTI Linear Time-Invariant

LPV Linear Parameter Varying

LTV Linear Time Varying

LMI Linear Matrix Inequalities

RG Reference Governor

QP Quadratic Programming

N4SID Numerical algorithm for space-state System IDentification

UAV Unmanned Aerial Vehicle

SV Surface vehicle

VTOL Vertical Take-Off and Landing

VMC Voltage Mode Control

CMC Current Mode Control

RP Rapid Prototyping

SISO Single-Input Single-Output

MIMO Multi-Input Multi-Output

NaSam2dB Nanorobotic-assisted Scanning Microwave Microscopy to analyze and manipulate 2D-and-Biomaterials

ADELE Aerial Drone for EnvironmentaL and Energy

LQR Linear Quadratic Regulator

LQG Linear Quadratic Gaussian

PID Proportional-Integral-Derivative

KF Kalman Filter

FT Fault-Tolerant

ROS Robot Operating System

Contents

List of Figures

List of Tables

Chapter 1

Summary of Ph.D. Research Activities

The aim of this chapter is to present the main activities that I performed during my Ph.D. in the triennium 2014-2017, at the Department of Information Engineering (DII) of the University "Politecnica delle Marche" (UNIVPM). The thesis is mainly focused on the design and development of model-based predictive controllers for electrical and mechatronic systems, giving theoretical advances and experimental results for the control of this kind of systems.

Model Predictive Control (MPC) is an advanced control technique that was proposed in the process industry already in the late sixties [2,3], and received a lot of attention since then [4,5]. What makes MPC successful in many engineering fields is the capability to optimally handle constrained multivariable systems, by making explicit use of a prediction model of the system. MPC derives the input sequence by solving a finite-time, open-loop, optimal control problem, and applies only the inputs corresponding to the actual time step, discarding the rest of the input sequence. This procedure is then repeated at each time step, in the so- called receding horizon fashion. Despite its long history, MPC is still one of the most flourishing fields in the control community, [6], and embedded MPC has seen an ever growing interest in the very last years [7,8]. Automotive, aerospace and electrical power systems are just some of the areas where the research in embedded MPC is very active, and these fields are particularly pushing towards more and more computationally efficient algorithms [9–12]. The necessity to solve online an optimization problem makes the computational burden of MPC higher than many other control algorithms, therefore its application to embedded systems is still a challenge. Indeed, the fast sampling frequency typical of embedded control is usually paired with reduced computational power, imposing severe limits to the diffusion of embedded MPC.

This thesis proposes MPC algorithms oriented to address several problems. The concept of MPC for pre-compensated systems is introduced for the control of power supplies, with an hard link to Reference Governor (RG) tech-

nique [13, 14], which has been already successfully used in other engineering fields, e.g. automotive and robotics [15–19]. The proposed algorithms, oriented to the implementation on embedded boards, give the theoretical conditions to effectively apply MPC on low computational power hardware of power supplies. The experimental results show the reliability and feasibility of the method for those embedded systems where the primal controller cannot be changed for several reasons, i.e. hard-coded/hardware-based algorithm.

Algorithms to control micro-positioning piezoelectric drivers have been also studied and developed during the activities related to the Nanorobotic-assisted Scanning Microwave Microscopy (NaSam2dB) research project. The NaSam2dB is a MIUR-DAAD Joint Mobilty Program, and the scientific objective of this cooperative project is to integrate a scanning microscope designed by UNIVPM into the nanorobotic characterization and manipulation framework developed by the german group at Oldenburg University (Division of Micro-robotics and Control Engineering - AMiR). In order to improve control performances of commercial micro-nano positioning systems, different model-based controllers have been designed and developed to improve transient response performance of the controlled system. To design the MPC, the piezoelectric system has been modeled in the form of a Linear Time Invariant (LTI) system. The developed control laws have given successfully results on different embedded control boards.

MPC algorithms for Unmanned Aerial Vehicles (UAV) have been developed during the activities within the Aerial Drone for EnvironmentaL and Energy (ADELE) research project. The ADELE is a research project funded by UNIVPM, and the scientific objective is to design and realize a "flying wing" drone equipped with electric propulsion propellers and hovering auxiliary control system for in-flight activities. The remotely piloted aerial vehicle will operate as an aerial platform to support analysis and data acquisition systems. Different MPC based on Linear Parameter Varying (LPV) and LTI description of the nonlinear UAV have been designed and developed in order to optimally drive the autonomous systems. Furthermore, an innovative solution has been developed to reduce the LPV-MPC computational effort. This solution uses a coordinate transformation to reduce the computational burden required by the MPC-LPV algorithm.This theoretical results is of paramount importance for the application of embedded LPV-MPC implemented on resource-constrained platform.

The research activities on autonomous vehicles have also considered a Fault Tolerant (FT) policy based on a MPC algorithm for the control of vessels. In this framework an algorithm has been designed to reduce the computational effort required to formulate the MPC control law with respect to the considered faults.

For the rapid prototyping of control and localization solutions on autonomous vehicles the use of the Robot Operating System (ROS) [20] has been considered as a flexible framework to write software for autonomous robotic platforms.

From the above research activities, the following publications have been obtained:

- Journal
  - Cavanini, L., Cimini, G., Ippoliti, G., and Bemporad, A. *"Model predictive control for pre-compensated voltage mode controlled DC-DC converters."* IET Control Theory & Applications (2017).
  - Cavanini, L., Cimini, G., Ippoliti, G. *"Model predictive control for pre-compensated power converters: application to current mode control."* IET Control Theory & Applications - under review
  - Cavanini Luca and Ippoliti G. *"Fault Tolerant Model Predictive Control for an Over-actuated Vessel."* Ocean Engineering - under review
  - Cavanini Luca, Ippoliti G. and Camacho E.F. *"Model Predictive Control of a Linear Parameter Varying Model of an UAV."* Journal of Guidance, Control and Dynamics - to be submitted to IEEE Transactions on Aerospace and Electronic Systems
  - Cavanini Luca, Ippoliti G. and Orlando G. *"Reference Governor of a Vertical Take-Off and Landing Fly-Wing Aircraft."* Journal of Guidance, Control and Dynamics - under review
  - Cavanini Luca, Cimini G., Ippoliti G. *"Computationally efficient model predictive control for a class of linear parameter-varying systems."* IET Control Theory & Applications - under review
- Conference Proceeding
  - Cavanini, L., Cimini, G., and Ippoliti, G. (2016, July). *"Model predictive control for the reference regulation of current mode controlled DC-DC converters."* In IEEE 14th International Conference on Industrial Informatics (INDIN), 2016 (pp. 74-79).
  - Cavanini, Luca, et al. *"Robust control of piezostage for nanoscale three-dimensional images acquisition."* In 42nd Annual Conference

of the IEEE Industrial Electronics Society, IECON 2016.

- Cavanini, L., Ciabattoni, L., Ferracuti, F., Ippoliti, G., and Longhi, S. (2016, July). *"Microgrid sizing via profit maximization: a population based optimization approach."* In IEEE 14th International Conference on Industrial Informatics (INDIN), 2016 (pp. 663-668).

- Cavanini Luca, Colombo L., Ippoliti G. and Orlando G. *"Development and experimental validation of a LQG control for a precompensated multi-axis piezo system."* In IEEE 26th International Symposium on Industrial Electronics (ISIE), 2017

- Cavanini Luca, Corradini M.L., Ippoliti G. and Orlando G. *"A model predictive control for a multi-axis piezo system: development and experimental validation."*In IEEE 4th International Conference on Control, Decision and Information Technologies (CoDIT), 2017.

- Cavanini Luca, Cimini G. and Ippoliti G. *"A fast model predictive control algorithm for linear parameter varying systems with rightinvertible input matrix."* In IEEE 25th Mediterranean Conference on Control and Automation (MED), 2017.

- Cavanini, L., Benetazzo, F., Freddi, A., Longhi, S. and Monteriú, A. *"SLAM-based autonomous wheelchair navigation system for AAL scenarios."* Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on. IEEE, 2014.

- Prist, M., Cavanini, L., Longhi, S., Monteriú, A., Ortenzi, D. and Freddi, A. *"A low cost mobile platform for educational robotic applications."* Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on. IEEE, 2014.

- Cavanini, L., Cimini, G., Ferracuti, F., Freddi, A., Ippoliti, G., Monteriú, A. and Verdini, F. *"A QR-code Localization System for Mobile Robots: Application to Smart Wheelchairs"* In IEEE 8th European Conference on Mobile Robotics (ECMR), 2017.

- Book Chapter

– Cavanini, L., Cimini, G., Freddi, A., Ippoliti, G. and Monteriú, A. *"rapros: a ROS Package for Rapid Prototyping."* Robot Operating System (ROS). Springer International Publishing, 2016. 491-508.s

1.1 Structure of the Thesis

This thesis is organized as follows.

Chapter 2 presents the notation used throughout the thesis, and some mathematical preliminaries.

In Chapter 3 the linear MPC problem and the corresponding Quadratic Programming (QP) formulation are described, presenting the considered control law formulation for LTI and LPV models, and finally the Reference Governor (RG) paradigm is given.

Chapter 4 reports the developed solutions for Linear Time Invariant (LTI) models. Controllers for power converters, piezoelectric actuators systems and LTI models of unmanned robotic systems have been considered. In the proposed solutions both mathematical modeling and identification results have been used to design the considered model-based control solutions.

Chapter 5 reports the developed solutions for Linear Parameter Varying (LPV) models. In particular two algorithms have been proposed, to control an autonomous aircraft named Aerosonde, which have been identified in the form of a LPV models. Theoretical aspects of an innovative solution to reduce the LPV-MPC computational effort for embedded implementation have been also detailed in this chapter.

Chapter 6 concludes the thesis, with a summary of what has been presented together with possible future developments of these research activities.

Chapter 2

Preliminaries

For the ease of the reader, this chapter goes over some basic mathematical preliminaries that will be used in the rest of the thesis. For a detailed discussion on them, please refer to textbooks on convex optimization [21,22].

2.1 Notation

Let $\mathbb{R}^n$ denote the set of real vectors of dimension n and $\mathbb{N}$ the set of natural integers, respectively. Let $\mathcal{I} \subset \mathbb{N}$ be a finite set of integers. For a vector $v \in \mathbb{R}^n$, v_i denotes the i-th entry of v. For a matrix $A \in \mathbb{R}^{n \times m}$, A' denotes its transpose. We denote [ ] the empty matrix ($n = 0$ or $m = 0$). For a square matrix $A \in \mathbb{R}^{n \times n}$, A^{-1} denotes its inverse, if it exists. Given a function $f(x) : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{dom} f$ denotes its domain.

2.2 Definitions

Definition 2.2.1. *(Convex set). A set $S \in \mathbb{R}^n$ is convex if for any x_1, $x_2 \in S$, and any ξ such that $0 \leq \xi \leq 1$, the following equation holds:*

$$\xi x_1 + (1 - \xi) x_2 \in S \tag{2.1}$$

that is, the line segment between any two points of the set belongs to the set itself.

Definition 2.2.2. *(Affine set). A set $S \in \mathbb{R}^n$ is affine if for any x_1, $x_2 \in S$, and any $\xi \in \mathbb{R}$, the following equation holds:*

$$\xi x_1 + (1 - \xi) x_2 \in S \tag{2.2}$$

that is, the line through any two points of the set belongs to the set itself. Every affine set is also convex.

Definition 2.2.3. *(Polyhedron). A polyhedron is the set of solution $x \in \mathbb{R}^n$ to a system of linear equalities and inequalities, such that*

$$\mathbb{P} = \{x | Gx \leq b, G_e x = b_e\} \tag{2.3}$$

with $G \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $G_e \in \mathbb{R}^{m_e \times n}$ and $b_e \in \mathbb{R}^{m_e}$. Polyhedra are convex sets.

Definition 2.2.4. *(Convex function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if $\boldsymbol{dom} f$ is a convex set and for any x_1, $x_2 \in \boldsymbol{dom} f$ and any ξ such that $0 \leq \xi \leq 1$, the following equation holds:*

$$f(\xi x_1 + (1 - \xi)x_2) \leq \xi f(x_1) + (1 - \xi) f(x_2.) \tag{2.4}$$

If $-f$ is convex, then f is concave.

Definition 2.2.5. *(Affine function). A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is affine if it is of the form $f(x) = Gx + b$, with $G \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. All affine functions are both convex and concave.*

Definition 2.2.6. *(Quadratic function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is quadratic if it can be written in the form:*

$$f(x) : \frac{1}{2} x' H x + h' x + k \tag{2.5}$$

with $H \in \mathbb{R}^{n \times n}$ a symmetric matrix, $h \in \mathbb{R}^n$ and $k \in \mathbb{R}$. A quadratic function is convex if and only if $H \geq 0$. In the rest of the thesis $k = 0$ holds, therefore the constant term k will be omitted.

Definition 2.2.7. *(Convex optimization). A convex optimization problem is of the form*

$$\begin{aligned} &min_x\ f_0(x) \\ &\quad s.t.\ f_i(x) \leq b_i,\ \ i = 1, \ldots, m, \end{aligned} \tag{2.6}$$

where $f_0, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex functions.

Definition 2.2.8. *(Quadratic programming). A quadratic programming (QP) problem minimizes a convex quadratic function over a polyhedron. A QP problem can be written in the following form:*

$$\begin{aligned} min_x\ &\frac{1}{2} x' H x + h' x \\ s.t.\ &Gx \leq b \\ &G_e x = b_e \end{aligned} \tag{2.7}$$

where $H \in \mathbb{R}^{n \times n}$ *is symmetric,* $H \geq 0$, $h \in \mathbb{R}^n$, $G \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $G_e \in \mathbb{R}^{m_e \times n}$ *and* $b_e \in \mathbb{R}^{m_e}$.

Chapter 3

Model Predictive Control

Model Predictive Control (MPC) is an advanced model-based control techniques allowing to deal with safety constraints and equipments routinely [4,5]. MPC groups a large set of predictive policies that consider a process model to compute an optimal control action by minimizing a cost function. Basically, the different predictive control policies share a set of common features:

- a mathematical model of the controlled process is considered to predict the system behavior over a time horizon;
- a cost function is minimized to compute the control signal to optimally drive the considered plant;
- the optimization sequence is computed iteratively with respect to the instantaneous system state and only the first predicted control effort is applied, according with the receding horizon strategy.

If the ability to explicit introduce constraints in the control law represents the main advantage provided by the MPC, there are also various issues that are directly addressed by this control policy: it can be used to drive complex multivariable dynamics by an intuitive tuning, it intrinsically compensates dead times and considers measured disturbances and the control law presents important additive features, as the possibility to consider the future reference when it is know a-priori. On the other hand, the main problem is still represented by the computational effort required to compute the MPC control law. Infact the predictive control is an optimal control law that requires to solve iteratively an optimization problem in the prescribed sampling time, and this aspect limits the application of MPC to slow dynamics processes. In order to overcome this problem, several solution have been proposed, i.e. unconstrained predictive controllers (equivalent to LQR and LQG controllers) [23], linear matrix inequalities (LMI) [24] and explicit approximated solutions (EMPC) [25]. In the last years, based on these approaches, and thanks to the improvement of hardware computational capabilities, MPC has found application to control nonlinear plants with fast dynamics. In this chapter different formulations

of the predictive control used in this thesis has been reported. For a comprehensive overview on the recent achievements and future developments of MPC, please refer to the survey paper [6]. In Sec. 3.1 the MPC minimization problem for discrete LTI models has been considered, reporting the condensed QP formulation (Sec. 3.1.1) and the standard state observer (Sec.3.1.2). The LPV-MPC approach has been considered in Section 3.2, reporting the LPV paradigm and the predictive control approach. In Section 3.3 the Reference Governor (RG) control approach based on predictive control has been reported.

3.1 Linear Time Invariant Model Predictive Control

In this section the MPC control law for LTI models is given, reporting cost function and constrains formulation and the condensed QP formulation considered in this thesis [4,5].

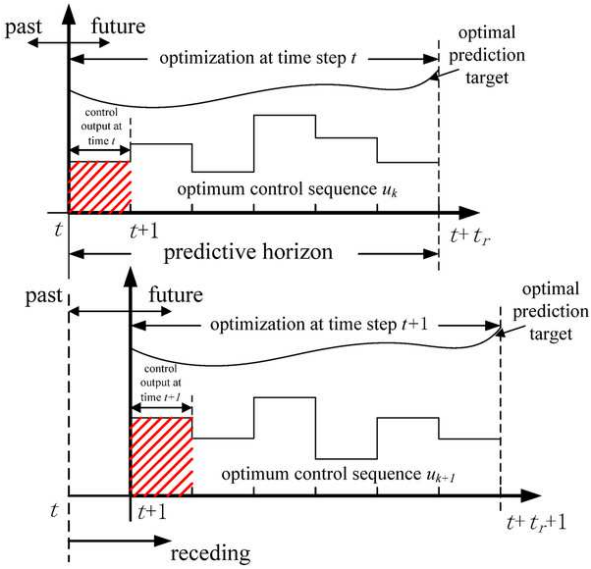

Figure 3.1: The open-loop optimization problem solved by MPC at two successive time sampling instant t and $t+1$.

The linear MPC consider a LTI representation of a real plant to predict the future system behavior and compute the optimal control sequence under prescribed constraints on input and output. A discrete time LTI plant model in space-state representation has been considered

$$\begin{aligned} x_{k+1} &= A_d x_{d_k} + B_d u_k \\ y_k &= C_d x_{d_k} \end{aligned} \tag{3.1}$$

where the system matrices are $A_d \in \mathbb{R}^{n_{x_d} \times n_{x_d}}$, $B_d \in \mathbb{R}^{n_{x_d} \times n_u}$ and $C_d \in \mathbb{R}^{n_y \times n_{x_d}}$, the state vector is $x_{d_k} \in \mathbb{R}^{n_{x_d}}$, the input vector is $u_k \in \mathbb{R}^{n_u}$ and the output vector is $y_k \in \mathbb{R}^{n_y}$. The objective of the controller is to steer the outputs y to track the references r.

The set of future inputs is computed by optimizing a cost function of the control performances, such as the tracking error and the control effort. The cost function is minimized subject to the constraints imposed by the dynamics of the system, i.e. the LTI model of Eq. (3.1), and constraints on both inputs and outputs. Only the inputs at time-step k are actually applied to the system, discarding the rest of the sequence. The procedure is repeated again at the next sampling time, shifting the horizon by one step (receding horizon). The number of future steps for which MPC computes the prediction at time k, is called prediction horizon and it is denoted by N_p. On the other hand, the number of "free moves" of the inputs to be optimized is called control horizon, denoted by N_u, and it is chosen such that $0 \leq N_u \leq N_p$ For the prediction instants greater then N_u, the last free move is "frozen" to the past value. This is done to reduce the computational effort required to solve the optimization problem, because the size of the primal vector depends on N_u. Figure 3.1 shows the operation of MPC at two successive time instances. Furthermore the constrained optimization problem solved iteratively by the MPC is

$$\min_{\Delta u} \quad \sum_{i=1}^{N_p-1} \|Q(y_{k+i|k} - r_k)\|_2^2 \tag{3.2a}$$

$$+ \sum_{j=0}^{N_u-1} \|R\Delta u_{k+j|k}\|_2^2 + \|P(y_{k+N_p|k} - r_{k+N_p})\|_2^2 \tag{3.2b}$$

$$\text{s.t.} \quad x_{d_{k+i+1|k}} = A_d x_{d_{k+i|k}} + B_d u_{k+i|k} \tag{3.2c}$$

$$y_{k+i+1|k} = C_d x_{d_{k+i+1|k}} \tag{3.2d}$$

$$x_{d_{k|k}} = x_{d_k} \tag{3.2e}$$

$$\Delta u_{k+N_u+j|k} = 0 \tag{3.2f}$$

$$\Delta u_{k+v|k} \in \mathbb{D} \tag{3.2g}$$

$$u_{k+v|k} \in \mathbb{U} \tag{3.2h}$$

$$y_{k+i|k} \in \mathbb{Y} \tag{3.2i}$$

$$i \in \{0, \ldots, N_p - 1\} \tag{3.2j}$$

$$j \in \{0, \ldots, N_u - 1\} \tag{3.2k}$$

where Q, R and P are weight matrices of appropriate dimension, $x_{d_{k+i|k}}$ denotes the prediction of the variable x_d at time $k+i$ based on the information available at time k, $\Delta u_{k+i|k}$ is the vector of the input increments, with $u_{k-1|k} = u_{k-1}$, r_k

is the vector of output references, $\mathbb{U}$, $\mathbb{D}$ and $\mathbb{Y}$ are polyhedral sets of constraints on inputs, input rate and outputs, respectively.

3.1.1 Condensed form

In this section the parametric condensed Quadratic Programming (QP) form of the MPC optimization problem (3.2) is presented. Defining $\Delta u_k = u_k - u_{k-1}$, the system of Eq. (3.1) became

$$\begin{aligned} \underbrace{\begin{bmatrix} \Delta x_{d_{k+1}} \\ y_{k+1} \end{bmatrix}}_{x_{k+1}} &= \underbrace{\begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \Delta x_{d_k} \\ y_k \end{bmatrix}}_{x_k} + \underbrace{\begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}}_{B} \Delta u_k \\ y_k &= \underbrace{\begin{bmatrix} 0 & I \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} \Delta x_{d_k} \\ y_k \end{bmatrix}}_{x_k} \end{aligned} \tag{3.3}$$

with $0 \in \mathbb{R}^{n_y \times n_{x_d}}$ and $I \in \mathbb{R}^{n_y \times n_y}$. Considering a sequence of N_u input for the plant model of Eq. (3.3), the predicted evolution of the plant over an horizon of N_p (such that $N_p \geq N_u$) time instances can be computed by

$$Y = Ex_k + \Phi \Delta U \tag{3.4}$$

where

$$\Delta U = \begin{bmatrix} \Delta u_k & \Delta u_{k+1} & \dots & \Delta u_{k+N_u-1} \end{bmatrix}'$$

$$Y = \begin{bmatrix} y_{k+1|k} + \dots + y_{k+N_p|k} \end{bmatrix}' \tag{3.5}$$

$$E = \begin{bmatrix} CA \\ CA^2 \\ \dots \\ CA^{N_p} \end{bmatrix}, \quad \Phi = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \dots & \dots & \dots & \\ CA^{N_p-1}B & \dots & \dots & CA^{N_p-N_u}B \end{bmatrix}. \tag{3.6}$$

The solution of the MPC optimization problem is the sequence of inputs ΔU that optimally drive the plant of Eq. (3.1). Considering prediction equation (3.4), the MPC cost function $J(x(k), \Delta U)$ can be formulated in the condensed form

$$J(x(k), \Delta U) = \Delta U' \underbrace{(\Phi' \bar{Q} \Phi + \bar{R})}_{H} \Delta U + \Delta U' \underbrace{\Phi' \bar{Q} (E x_k - S)}_{h} \tag{3.7}$$

where, from Eq. (3.2) the weight matrices are

$$\bar{Q} = \begin{bmatrix} Q^2 & 0 & \dots \\ \dots & \dots & \dots \\ \dots & Q^2 & \dots \\ \dots & 0 & P^2 \end{bmatrix} \quad , \quad \bar{R} = \begin{bmatrix} R^2 & 0 & \dots \\ \dots & \dots & \dots \\ \dots & R^2 & \dots \\ \dots & 0 & R^2 \end{bmatrix} \tag{3.8}$$

and, considering $u^{(i)m}$ the input bounds values at the prediction time instance i, $\Delta u^{(i)m}$ the input rate bounds values at the prediction time instance i and $y^{(j)m}$ the output bounds at the prediction time instance j, with i=0, …, N_u-1 and j=0, …, N_p-1, the bounds matrices in condensed form are

$$\begin{aligned} U_{min} &= \begin{bmatrix} u^{(0)m} \\ \vdots \\ u^{(N_u-1)m} \end{bmatrix} , \; U_{max} = \begin{bmatrix} u^{(0)M} \\ \vdots \\ u^{(N_u-1)M} \end{bmatrix} , \\ \Delta U_{min} &= \begin{bmatrix} \Delta u^{(0)m} \\ \vdots \\ \Delta u^{(N_u-1)m} \end{bmatrix} , \; \Delta U_{max} = \begin{bmatrix} \Delta u^{(0)M} \\ \vdots \\ \Delta u^{(N_u-1)M} \end{bmatrix} , \\ Y_{min} &= \begin{bmatrix} y^{(0)m} \\ \vdots \\ y^{(N_p-1)m} \end{bmatrix} , \; Y_{max} = \begin{bmatrix} y^{(0)M} \\ \vdots \\ y^{(N_p-1)M} \end{bmatrix} \end{aligned} \tag{3.9}$$

such that the bounds matrix S is

$$S = \begin{bmatrix} U_{max} & -U_{min} & \Delta U_{max} & -\Delta U_{min} & Y_{max} & -Y_{min} \end{bmatrix}' . \tag{3.10}$$

In Eq. (3.7) the hessian matrix $H \in \mathbb{R}^{n_z \times n_z}$ is diagonal and semi-definite positive and the matrix $h \in \mathbb{R}^{n_z}$ is the linear term. The problem of Eq. (3.7) is a convex optimization problem and the solution in absence of constraints can be obtained in closed form, equalizing to zero the first derivative of the cost function. Considering a parameters array $\rho_k = \begin{bmatrix} u_{k-1} & x_k & r_k \end{bmatrix}' \in \mathbb{P}$ used to update iteratively the problem with respect to the plant state at time instance k where $\mathbb{P} \in \mathbb{R}^{n_\rho}$ is a bounded set of interest, the Eq. (3.7) defines the QP problem

$$\min_z \quad \frac{1}{2} z' H z + \rho_k' F' z \tag{3.11}$$

where $F \in \mathbb{R}^{n_z \times n_\rho}$ and $z \in \mathbb{R}^{n_z}$ is the optimization variables array.

The introduction of constraints in the minimization problem allows to consider more specifics directly in the control law. Standard MPC constraints allow to impose bounds to the generated control signal, the variation rates and

the plant output values. Constraints are formulated in terms of ΔU of Eq. (3.7) in order to obtain a parametric compact formulation as

$$\begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \Delta U \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{3.12}$$

where

$$\begin{aligned} G_1 &= \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}, \quad b_1 = \begin{bmatrix} -U_{min} + C_1 u_{k-1} \\ U_{max} - C_1 u_{k-1} \end{bmatrix} \\ G_2 &= \begin{bmatrix} -I \\ I \end{bmatrix}, \quad b_2 = \begin{bmatrix} -\Delta U_{min} \\ \Delta U_{max} \end{bmatrix} \\ G_3 &= \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}, \quad b_3 = \begin{bmatrix} -Y_{min} + Ex_k \\ Y_{max} - Ex_k \end{bmatrix} \end{aligned} \tag{3.13}$$

with C_1, C_2 having the same size of F, Φ and structure

$$C_1 = \begin{bmatrix} I \\ I \\ \dots \\ I \end{bmatrix}, \quad C_2 = \begin{bmatrix} I & 0 & \dots & \dots & 0 \\ I & I & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ I & I & \dots & I & I \end{bmatrix}. \tag{3.14}$$

Considering the QP problem formulation given in Eq. (4.28), constraints on input rate, input and output of Eq. (3.12), can be expressed with respect to the parameter array ρ_k of Eq. (4.28) in order to formulate the constrained QP problem of MPC as

$$\min_z \ \frac{1}{2} z' H z + \rho_k' F' z \tag{3.15}$$

$$\text{s.t. } Gz \leq W\rho_k + w_k \tag{3.16}$$

with $G \in \mathbb{R}^{m \times n_z}$, $W \in \mathbb{R}^{m \times n_\rho}$ and $w_k \in \mathbb{R}^m$. Constraint formulation given in Eq. (3.12) considers to limit the input rate, the input and the output maximum and minimum value, in order to avoid unrequested behavior related to actuators saturations and process physical limits.

MPC constraints are classified in hard and soft constraints:

- **hard constraints** must be satisfied at all times otherwise the problem is infeasible
- **soft constraints** can be violated to avoid infeasibility. Strategies for handling soft constraints are *i*) remove least critical constraint until optimization is feasible or *ii*) impose hard constraints on the probability of violating each soft constraint.

In order to prevent infeasibility of the QP problem, the standard approach is represented by the introduction of a panic (slack) variable in the optimization problem [4]. The slack variable, related to a weight too big with respect to other optimization variables, allows to expand the space of solutions, widening problem constraints in the considered optimization. The conditions for the stability of this control scheme have been formalized in [26], where it is shown that nominal stability of MPC for linear systems is achieved either by adding a terminal cost and constraint, or by extending the prediction horizon.

3.1.2 State observer

MPC uses the instantaneous state value to compute iteratively process evolution with respect to the optimization sequence. In real world processes, due to the presence of disturbances, noise and model uncertain, an appropriate filter is required in order to provided an estimation of the state array closed to the real one. Furthermore, prediction model can not correspond to the real process model. In order to obtain the required plant state array estimation, usually a Kalman state estimator is introduced in the MPC control law. Kalman filter has been proposed in 1960 from R.E. Kalman in his paper describing a recursive solution to the discrete time linear filtering problem [27]. The discrete Kalman filter, formulated with respect to the discrete LTI model of Eq. (3.1) is composed by the well known equations:

$$\begin{aligned} \hat{x}_{d_{k|k}} &= \hat{x}_{d_{k|k-1}} + M(y_k - C_d \hat{x}_{d_{k|k-1}}) \\ \hat{x}_{d_{k+1|k}} &= A_d \hat{x}_{d_{k|k-1}} + B_d u_k + L(y_k - C_d \hat{x}_{d_{k|k-1}}). \end{aligned} \tag{3.17}$$

3.2 Linear Parameter-Varying Model Predictive Control

Nonlinear Model Predictive Control (NL-MPC) is considered one of the main solution for real-time control of nonlinear processes and it find application in a large range of fields. On the other hand, NL-MPC requires a strong computational burden that reduce the set of real world applications to the slower control problem. Nowadays, a standard solution to optimal constrained control of nonlinear plants considers to describe processes by a Linear Parameter-Varying (LPV) models. These models are based on linear or piecewise-linear approximations of the nonlinear plant.

Linear Parameter-Varying (LPV) formalism has been introduced by Shamma in his Ph.D. thesis [28] in 1988 considering the design of scheduled controllers. Gain scheduling control techniques consider the control of nonlinear systems by nonlinear controllers designed patching together a set of linear controllers

blended on-line with the plant evolution by interpolation or switching policies. LPV representation is widely used to describe not only nonlinear plants but also several dynamics, such hybrid dynamical systems (the parameters belong to a set of discrete values) or jump linear systems (the evolution of the parameters can be described by probabilistic rules). LPV considers a continuous nonlinear plant in space-state form

$$\begin{aligned} \dot{x}(t) =& A(t)x(t) + B(t)u((t) \\ y(t) =& C(t)x(t) \end{aligned} \tag{3.18}$$

with $A(t)$, $B(t)$ and $C(t)$ time varying matrices evolving with some nonlinear law. In the LPV framework the nonlinear plant of Eq. (3.18) is represented by a dynamical model as

$$\begin{aligned} \dot{x}(t) =& A(\theta(t))x(t) + B(\theta(t))u((t) \\ y(t) =& C(\theta(t))x(t) \end{aligned} \tag{3.19}$$

where $\theta(t)$ is an array of time-varying *scheduling parameters* allowing to describe the nonlinear system behavior by a collection of indexed LTI models. The parameter array is named *exogenous* when is not related to the state evolution and *endogenous* when it is possible to represent it evolution by some state variable relationship $\theta(t) = h(x(t))$.

LPV approach allows to abstract away this nonlinear dependency, resulting in linear, but non-stationary, dynamics [29]. LPV framework is strictly related to the LTI paradigm as much as the difference between two approaches is evident (LTI models are stationary, LPV representation are not) [30]. A less obvious difference distinct LPV and Linear Time Varying (LTV) models, that differ for considerations about stability and controller synthesis.

In order to obtain a suitable representation of a LPV model as in Eq. (3.18), several representations of time-varying matrices have been proposed:

- Affine parameters dependence of linear parameter-varying models (LPV-A) [31] considers to explicit a generic $L(t) = L(\theta(t))$ matrices in the form

  $$L(t) = L_0 + L_1\theta_1(t) + L_2\theta_2(t) + \cdots + L_{n_\theta}\theta_{n_\theta}(t) \tag{3.20}$$

  that can be particularized in input affine parameter dependence (LPV-IA), where B, D matrices are time varying and C, A matrices are constant.

- Rational parameter dependance representation (LPV-R) [32] consider to

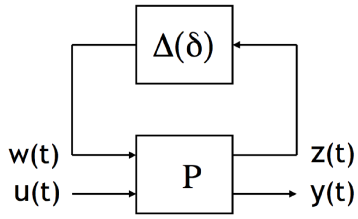

Figure 3.2: Linear fractional representation of a linear parameter-varying model

describe the time varying matrix as

$$L(t) = \Big[L_{n0} + L_{n1}\theta_1(t) + \cdots + L_{nn_\theta}\theta_{n_\theta}(t)\Big] \tag{3.21}$$

$$\Big[I + L_{d1}\theta_1(t) + \cdots + L_{dn_\theta}\theta_{n_\theta}(t)\Big]^{-1} \tag{3.22}$$

where L_{ni} and L_{di} are the denominator and numerator of the term related to the parameters θ_i.

- Linear fractional representation parameter dependence (LPV-LFR) [33] considers to represent a space-state LVP model in the form

$$\begin{aligned}
\dot{x}(t) =& Ax(t) + B_0 w(t) + B_1 u(t) \\
z(t) =& C_0 x(t) + D_{00} w(t) + D_{01} u(t) \\
y(t) =& C_1 x(t) + D_{10} w(t) + D_{11} u(t) \\
w(t) =& \Delta z(t) \\
\Delta =& \begin{bmatrix} \theta_1 I & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \theta_{n_\theta} \end{bmatrix}.
\end{aligned} \tag{3.23}$$

  The block representation is given in Fig. 3.2.

The above time-varying matrix representations have been used to develop different LPV modeling techniques, considering the standard LTI input-output representation of plant

$$y(t) = -\sum_{i=1}^{n_a} a_i(\theta(t))y(t-i) + \sum_{j=1}^{n_b} b_j(\theta(t))u(t-j) \tag{3.24}$$

and extending the system description of Eq. (3.24) by affine, rational or linear fractional approach to model nonlinear plant in the LPV framework [33, 34]. Furthermore, considering the real world control problems, in absence of a reliable mathematical of the plant, several LPV identification procedures have

been proposed in order to obtain a useful plant model for the control. These procedures are classified as [35]:

- Local approach [36]: by different experiments, a set of local LTI models is identified and the final LPV plant is computed interpolating time invariant systems along the parameter trajectory;
- Global approach [31]: consider a single experiment in which the parameter is excited to assume all values of interest in the studied scenario. A parameter-dependent model is directly obtained.

Both approaches can be used to solve the identification problem obtaining input-output representations as space-state LPV models. Considering local approach to the LPV identification, the main solutions are:

- Steinbuch approach [37] considers to estimate a transfer function from the local frequency response and then to interpolate a global function fitting the local ones. The global transfer function is finally casted in the Canonical Controllability Form (CCF). The issues is that the CCF conversion not guarantees the validity of the local models in the same state-space basis;
- Paijmans method [38] parametrizes local models by poles, zeros and gain factored in low order subsystems and decomposed according with a set of rules, but it is valid only for SISO systems;
- Lovera approach [39] addresses numerical issues identifying local discrete LTI models by subspace algorithms, balancing the models by numerical algorithm and using the bilinear transformation to obtain continuous time models. This approach can be used to obtain the Linear Fractional Representation (LFR).

The widely used global LPV identification techniques are:

- Nemani [40] solution considers to identify single LPV-LFR systems with a scalar parameter, with state array accessible, using root least square method;
- Lovera method [41] is oriented to MIMO plants considering a least square optimization problem, introducing noisy state vector measurement in the identification, if available;
- Lee and Poolla [42] technique is a maximum likelihood MIMO LPV-LFR plant identification solution, and the main issue is related to its initialization which is based on the computation of gradient and hessian;

- Subspace methods [43], related to the extension of the LTI subspace approach to LPV plants and proposed in several versions.

In this thesis, the LPV space state model representation is considered, assuming that the parameter array θ_k is iteratively measurable or estimable, such that the considered nonlinear process can be described iteratively by a LTI model. This LTI is obtained fixing the parameter array instantaneous value

$$\begin{aligned} x_{k+1} &= A(\theta_k)x_k + B(\theta_k)u_k \\ y_k &= C(\theta_k)x_k. \end{aligned} \tag{3.25}$$

The LPV model of Eq. (3.25), represented in terms of the instantaneous value of the time-varying parameter array θ_k, can be described by a space state model defined by the space state matrices evaluated at the time instance k, $A(\theta_k) = A_k$, $B(\theta_k) = B_k$ and $C(\theta_k) = C_k$, such as

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k \\ y_k &= C_k x_k. \end{aligned} \tag{3.26}$$

The standard LPV-MPC paradigm [44] considers to iteratively reformulate the QP problem with respect to θ_k, evaluating at each time instance the plant model of Eq. (3.26) [45], such as

$$\begin{aligned} \min_z \ & \frac{1}{2} z' H_k z + \rho_k' F_k' z \\ & G_k z \leq W_k \rho_k + w_k \end{aligned} \tag{3.27}$$

where $H_k = H(\theta_k)$, $F_k = F(\theta_k)$, $G_k = G(\theta_k)$ and $W_k = W(\theta_k)$.

3.3 Reference Governor

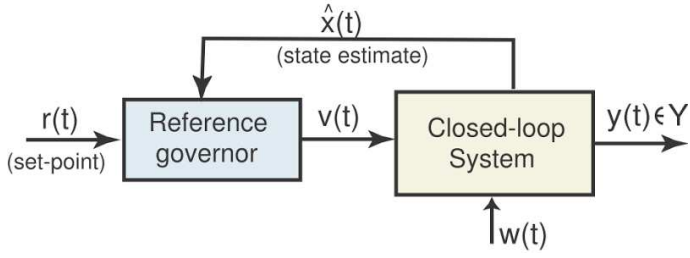

Figure 3.3: The Reference Governor control scheme

Reference and command governors are add-on control schemes, as shown in Fig. 3.3, which enforce state and control constraints on pre-stabilized systems by modifying, whenever necessary, the reference signal [16]. Model Predictive Control based Reference Governor (RG) policies consider the introduction of a slower external MPC loop developed with respect to the internal control loop model. The main advantage given by the introduction of the MPC loop is represented by the iterative optimization of the reference trajectory that also allows to avoid potential constraints violation [46].

3.3.1 Reference Governor for LTI system

In order to resume basic elements of RG policy for multi-input multi-output systems, usually named vector RG [16], the approach to LTI pre-compensated system is here briefly presented. When the vector RG policy is used to control single-output single-input processes, it is usually known as scalar RG.

A discrete multi-input multi-output pre-compensated LTI system in space state form is considered

$$\begin{aligned} x_{k+1} &= Ax_k + Bv_k + Bw_k \\ y_k &= Cx_k + Dv_k + D_w w_k \end{aligned} \tag{3.28}$$

with $x \in \mathbb{R}^{n_x}$ the state vector, $v \in \mathbb{R}^{n_v}$ the primal controller reference vector, $w \in \mathbb{R}^{n_w}$ the disturbance vector and $y \in \mathbb{R}^{n_y}$ the plant output vector. The disturbance vector is considered unmeasured but bounded, with know bounds such that $w \in \mathbb{W}$, where $\mathbb{W}$ is a given compact set. Due to the presence of the primal controller, the pre-compensated system is assumed asymptotically stable, and the state transition matrix A can be assumed to be a Schur matrix (all eigenvalues are in the interior of the unit disk). Constraints are imposed on the controlled output, such that

$$y_{k+i} \in \mathbb{Y} \ , \quad \text{for } i > 0 \tag{3.29}$$

where $\mathbb{Y} \in \mathbb{R}^{n_y}$, with $\mathbb{R}^{n_y}$ a prescribed set. For computational reason $\mathbb{Y}$ is a convex polytope [47]. The target of the RG is to drive the applied reference, v_k, is such a way that it is as close as possible to the desired reference r_k (unknown in advance), and the constraints are enforced. The vector RG consider the reference signal update law [48]

$$v_k = v_{k-1} + K_k \left(r_k - v_{k-1} \right) \tag{3.30}$$

where $K_k \in \mathbb{R}^{n_v}$ is the update matrix, considered diagonal to decouple the computed reference signal update. In order to iteratively compute optimally

the elements of the update matrix K_k, they are obtained minimizing a cost function

$$\min_{v_k}(v_k - r_k)Q(v_k - r_k)' \tag{3.31}$$

where $Q \in \mathbb{R}^{n_v}$ is a positive definite symmetric weight matrix ($Q = Q^{'} > 0$).

3.3.2 Reference Governor policies

Considering the proposed approach to the RG control problem, and its formulation in the form of a quadratic programming problem, MPC-based RG is become the standard approach to the command control problem. Furthermore, innovative formulations and RG solutions consider several approaches with respect to standard MPC feedback control:

- The feedforward reference governor idea is that if the set-point modifications are sufficiently slow then one can have high confidence on the expected value of the state, even in the absence of an explicit measurement of it, because of the asymptotic stability of the pre-compensated system at hand [49];
- Reduced order reference governors for systems with states decomposable into "slow" and "fast" states can be based on the reduced order model for "slow" states, provided constraints are tightened to ensure that the contributions of fast states do not cause constraint violation [50];
- Network reference governor handling variable delays, where the controller and the plant are connected via a, usually non-ideal, communication network [51];
- Virtual state governor for integrating existing controllers solves a slightly different problem, namely how to integrate multiple actuators, each one equipped with an assigned non-modifiable feedback control law [8].

The topic of these governor policies is to preserve the pre-compensated system, by ensuring that the modified reference command is closed to the original reference subject to constraints. The scalar reference governor is attractive as it leads to computationally simple implementations for both linear and nonlinear systems with disturbances and parameter uncertainties. Other reference governor policies are more complex but provide better performance or address special problems [46, 52]. For a complete review of the state of the art of RG it is possible to refer to [16].

Chapter 4

MPC for LTI systems

MPC for Linear Time Invariant (LTI) models represents the basic approach to the predictive control, as yet described in Chapter 3. Despite of the real world system are featured by nonlinear dynamics, LTI-MPC represents a valid approach to optimally control a wide set of real systems. In this chapter different MPC algorithms are proposed, presenting several approaches allowing to control the considered plants. In Sec. 4.1 MPC policies for the control of DC-DC buck power converters have been presented. The controlled converters performance are improve by the introduction of an external MPC loop that, according to the paradigm of the Reference Governor (RG) (see Sec. 3.3), optimally modifies the internal loop reference signal. Theoretical conditions to effectively apply MPC for pre-compensed systems on low computational hardware have been also given. In Sec. 4.2, considering both MPC and RG approaches, different control laws for piezoelectric actuators are proposed. In the considered approach the piezo systems are represented by a LTI model obtained by an identification procedure. Different MPC algorithms are proposed and the control results are reported, showing the goodness of the proposed controllers, in comparison with respect to provided commercial controllers. Section 4.3 presents LTI-MPC algorithms for the control of Unmanned Vehicles. In Section 4.3.1, an LTI-MPC for the control of a pre-compensated VTOL aircraft is proposed. The system, provided by a low level linear controller has been modeled in the form of an LTI system considering to trim it in an appropriate operating operating point. The RG formulated with respect to the considered LTI systems has been used to improve the control performances of the pre-compensated aircraft over all the possible asset configurations. The improvement has been compared with respect to the primal controllers' results. Section 4.3.2 concludes the chapter presenting a Fault Tolerant (FT) MPC algorithm to control an over-actuated vessel. In the proposed algorithm the vessel is modeled by an LTI system and by a feedback loop from the actuators and the controller, a FT policy has been introduced in the MPC. The proposed control law allows to improve the control performance in case of actuators' faults, and also to define non controllable scenarios. The FT-MPC algorithm has been compared

with standard MPC controller in order to show the performance improvement introduced by the proposed FT policy.

4.1 Power Converter

In the last years, the continuously increasing tightening of efficiency and performance requirements (ENERGY STAR®) enhanced several research fields in the control of power supplies [53]. Among them, Model Predictive Control (MPC) saw a wide interest from both academy and industry [54]. The main aspects that favored the interest of MPC in this field are the intuitive design and tuning of the controller, the enhanced performance and the availability of relatively accurate models for electrical devices [55]. Such highly requested features come at the cost of cumbersome online calculations, which have limited the spread of the method in fast-sampled systems. For these reasons, researchers have been encouraged in looking into efficient solutions for embedded MPC implementations [56], instead of using powerful embedded boards, like Field Programmable Arrays (FPGAs), that could not be available for such systems [57]. Moreover, the very recent complexity certification of Quadratic Programming (QP) solvers is a step forward into safe implementation of embedded MPC [56]. When dealing with transistors-based devices, the literature splits into two branches. *Continuous Control Set* (CCS)-MPC takes control actions into a continuous set, which is usually the duty cycle of the Pulse Width Modulation (PWM). *Finite Control Set* (FCS)-MPC exploits the discrete nature of power converters and takes action in a discrete control set namely the finite combinations of predicted switches' states [58]. To deal with the computational load, explicit MPC is the preferred solution for CCS-MPC. It presolves the optimization problem through multiparametric Quadratic Programming (mpQP), and the implemented controller turns to be a PieceWise Affine (PWA) function of parameters, easy to be applied online [59]. Explicit MPC has already been successfully used for power converters control [60]. However, it becomes impractical when the number of inputs or the prediction horizon is not small enough, due to the high memory requirements. On the other hand, several algorithms have been proposed to efficiently implement FCS-MPC, and to achieve long prediction horizons [58]. The performances of CCS and FCS-MPC have been recently compared [61]. FCS-MPC usually provides a faster response time than CCS-MPC. However, CCS-MPC decouples the switching frequency from the controller sampling time, and it operates the converter at a fixed frequency. For these reasons CCS-MPC is more often preferred in industrial applications. The aim of this section is to investigate CCS-MPC for pre-compensated DC-DC converters. Following the idea of the Reference Governor (RG)(see Sec. 3.3), this thesis presents the design of an MPC loop that

regulates a DC-DC converter while manipulating the reference of the actual primal controller [13, 14]. Several engineering fields have already experimented the use of RG, like automotive and robotics [15]. However, the focus of the current research on power electronics' control algorithms, is almost completely based on replacing the standard controller. Only few recent attempts can be found where the modification of the reference is used to improve the transient response of standard controllers [62]. Power conversion seems to be a field where the control of pre-compensated systems could find an important role. Indeed, very often there is no possibility to change the native controller, which is hard coded or even hardware based [63]. The designer could also have the necessity to retain the primal controller due to stability and robustness certification [64]. Furthermore a double loop, multi-rate, control structure would permit to exploit the benefits of MPC with a slower sampling frequency, improving the feasibility of CCS-MPC in such fast sampling and computationally cheap systems. In the following, the application of MPC to manage the reference of pre-compensated power converters have been tested, considering both Voltage Mode Control (VMC) and Current Mode Control (CMC) primal control strategies. This chapter is structured as follows.

In Sec. 4.1.1 the buck converter plant model has been considered. The proposed MPC-VMC and MPC-CMC control policies have been introduced in Sec.4.1.2 and 4.1.3, respectively, where the control laws formulation and the simulation and experimental control tests have been given.

4.1.1 DC-DC Buck Converter

DC-DC converters are switching systems due to the discrete input that assumes values in the set $\{0, 1\}$. This input corresponds to the on/off state of the switch. Although discrete actuation is a possible alternative, PWM converters are usually preferred in industrial applications. The main reason is the fixed switching frequency provided by the modulator which reduces the stress of the components. In addition, this permits to decouple switching and control frequencies, guaranteeing at the same time good ripple reduction and satisfactory sampling time. The model of PWM converters are averaged continuous model over a switching period [65]. For the DC-DC buck converter considered, the averaged model is:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.1a}$$

$$y(t) = Cx(t) \tag{4.1b}$$

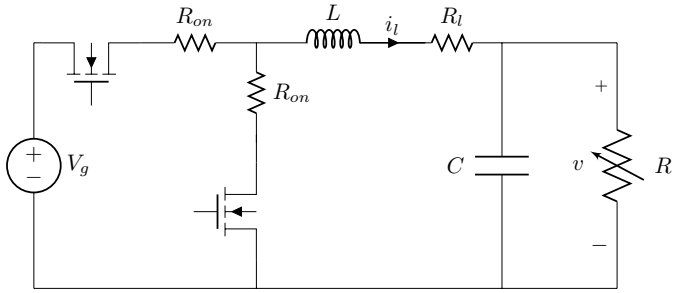

Figure 4.1: Electrical schematic of the synchronous DC-DC buck converter.

with $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$, and

$$A = \begin{bmatrix} -\frac{R_l + R_{on}}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix}, B = \begin{bmatrix} \frac{V_g}{L} \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}. \tag{4.2}$$

The states $x(t) = \begin{bmatrix} i_l(t) & v(t) \end{bmatrix}'$ are the inductor current and the output voltage, respectively, whereas the input $u(t) = d(t)$ is the duty cycle of the PWM and gets values in the range $\mathbb{R}_{[0,1]}$. L and C are the inductor and capacitor values, R_l and R_{on} the parasitic components of the inductor and switch, V_g is the input voltage, and R is the supplied load. Synchronous converters are usually controlled with a single input, with a master-slave technique that drives the two switches [65]. In buck converters the primary switch is high-side, and directly driven by the PWM signal. The second transistor is low-side, and actuated by a complementary signal.

4.1.2 MPC for pre-compensated VMC Buck Converter

Voltage based control must guarantee the regulation of the output voltage at the desired value. The most widely used controllers are Voltage Mode Control (VMC) and Current Mode Control (CMC), and nowadays they are considered a technology [65, 66]. The first has a single loop with voltage feedback, the second presents a cascaded structure with inner current and outer voltage control loops. CMC exploits the faster dynamics of the current to provide a responsive control and over current protection, however it needs a current sensor which brings extra cost, space and susceptibility to noise [66]. For these reasons, VMC is preferred in some applications for its simplicity. Nonetheless VMC and CMC are still subject of research for performance improvements [67]. However, standard linear control is the first choice for consumer electronics, and improvements in this field are not the aim of this thesis. The standard VMC

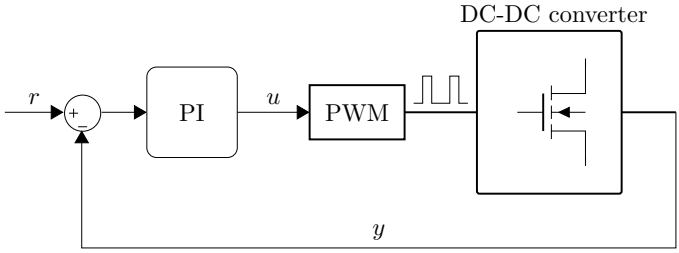

Figure 4.2: Standard Voltage Mode Control.

has been implemented with a linear Proportional Integral (PI) regulator [65]. The block scheme of the controller is depicted in Fig. 4.2. The parameters of the controller are tuned with small-signals modeling, see e.g. [68]. Bandwidth and stability margins are usually considered to obtain a satisfactory controller. As a rule of thumb, a controller that exhibits a gain margin of about 10dB and a phase margin greater than 45 degrees is desirable [65, 68]. The linear controller derived has been obtained following this general rule of thumb.

4.1.2.1 Closed-loop VMC model

In order to design the MPC controller, the mathematical model of the pre-compensated closed-loop system must be known, and it is derived in the following. The purpose of this section is to remain as general as possible, thus the controller derivation is carried out for a PID regulator, even if PI controllers usually guarantee satisfactory performance for DC-DC converters.

The discrete-time Linear Time Invariant (LTI) model of the system is directly obtained from equations (4.1) and (4.2), such as

$$x(k+1) = A_d x(k) + B_d u(k) \tag{4.3a}$$
$$y(k) = C_d x(k), \tag{4.3b}$$

where $k \in \mathbb{N}$ is the discrete-time index, $A_d = e^{AT_s}$, $B_d = \int_0^{T_s} e^{A\tau} d\tau B$, and $C_d = C$, are the discrete-time state-space matrices.

Definition 4.1.1. *Consider the parallel formulation of a PID controller*

$$u(k) = \left(K_p + K_i T_s \frac{z}{z-1} + K_d \frac{\delta}{1 + \delta T_s \frac{z}{z-1}} \right) e(k) \tag{4.4}$$

its discrete-time LTI state space representation is $\mathcal{S}_c(A_c, B_c, C_c, D_c)$*, with*

$$\begin{aligned} x_c(k+1) &= \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix}}_{A_p} x_c(k) + \underbrace{\begin{bmatrix} \tilde{K}_I \\ -\tilde{K}_D(1-\alpha) \end{bmatrix}}_{B_p} u_c(k) \\ y_c(k) &= \underbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}_{C_p} x_c(k) + \underbrace{\begin{bmatrix} K_P + \tilde{K}_I + \tilde{K}_D \end{bmatrix}}_{D_p} u_c(k) \end{aligned} \tag{4.5}$$

where

$$\tilde{K}_i = K_i T_s \quad , \quad \tilde{K}_d = \frac{K_d}{T_s + t_f} \quad , \quad \alpha = \frac{t_f}{t_f + T_s}. \tag{4.6}$$

One can trivially verify that $y_p(k) \equiv u(k)$ and $e(k) \equiv u_p(k) \equiv r(k) - y(k)$ where $r(k) \in \mathbb{R}^{n_y}$ is the reference signal for the controlled system, that is the output voltage reference. The subscript p will stand for *primal controller* model. Let $y_c(k) \equiv y(k)$ be the output of the extended system, and consider the extended state vector $x_c(k) \in \mathbb{R}^{n_{x_c}}$ defined as

$$x_c(k) = \begin{bmatrix} x_p(k) \\ x(k) \end{bmatrix}. \tag{4.7}$$

The extended open-loop system is therefore equal to

$$x_c(k+1) = \underbrace{\begin{bmatrix} A_p & 0 \\ B_d C_p & A_d \end{bmatrix}}_{A_c} x_c(k) + \underbrace{\begin{bmatrix} B_p \\ B_d D_p \end{bmatrix}}_{B_c} u_p(k) \tag{4.8a}$$

$$y_c(k) = \underbrace{\begin{bmatrix} D_d C_p & C_d \end{bmatrix}}_{C_c} x_c(k) + \underbrace{\begin{bmatrix} D_d D_p \end{bmatrix}}_{D_c} u_p(k), \tag{4.8b}$$

with the tracking error $e(k) \equiv u_p(k)$ as its only input. The closed-loop model derived from (4.8) is

$$x_c(k+1) = A_f x_c(k) + B_f r(k) \tag{4.9a}$$

$$y_c(k) = C_f x_c(k) + D_f r(k) \tag{4.9b}$$

where, by setting $d \triangleq (D_c + 1)^{-1}$, the following equations hold

$$A_f = A_c - B_c C_c d \tag{4.10a}$$

$$B_f = B_c - B_c D_c d \tag{4.10b}$$

$$C_f = C_c d \tag{4.10c}$$

$$D_f = D_c d. \tag{4.10d}$$

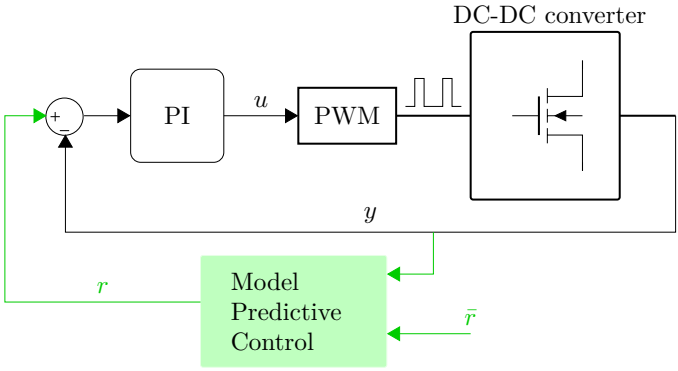

Figure 4.3: Proposed Model Predictive Control - Voltage Mode Control.

Model (4.9), with matrices defined as in (4.10), represents the closed-loop system of an LTI system controlled by a linear PID regulator. However, it is a common practice to use simple PI controllers for DC-DC converters as they are proven to give satisfactory performance for such systems [65]. This holds true also for the experimental results presented in this work, that is $K_d = 0$ in Eq.(4.4). With a standard controller the provided set-point and the reference to the primal controller are the same, namely $r = \bar{r}$. With MPC-VMC, the set-point is $\bar{r}$ and the dynamical reference to the primal controller is $r(k)$ (see figures 4.2 and 4.3).

4.1.2.2 MPC-VMC

In embedded MPC, the time required to solve the optimization problem is not negligible with respect to the sampling interval. Therefore, for a correct synchronization, the input computed by MPC is applied to the system with one-step delay. Consequently MPC regulates the following system:

$$x_c(k+1) = A_f^\eta x_c(k) + B_f^\eta \mathring{r}(k-1) \tag{4.11a}$$

$$y_c(k) = C_f^\eta x_c(k) + D_f^\eta \mathring{r}(k-1) \tag{4.11b}$$

with $\mathring{r}(k-1) = r(k)$, and A_f^η, B_f^η, C_f^η, D_f^η the state-space matrices of the discrete-time system (4.9) down sampled at $T_s^\eta = \eta T_s$, with $\eta \geq 1$. This formulation allows the MPC controller to eventually run at slower frequency with respect to the primal controller, for computational load reasons.

The considered quadratic cost function subject to the linear equality con-

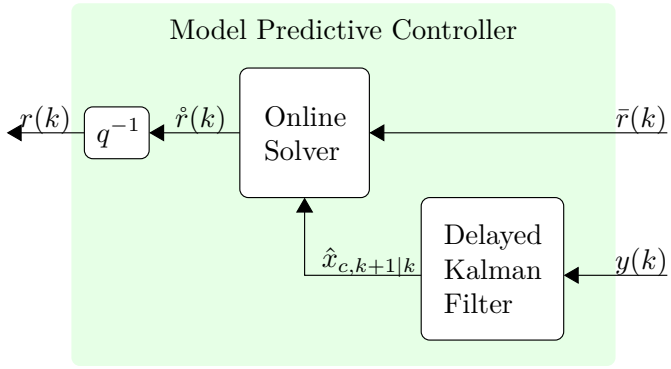

Figure 4.4: Block scheme of the MPC.

straints representing the model dynamics is:

$$\min_{\Delta\mathring{r}} \quad \sum_{i=1}^{N_p} \|Q(y_{c,k+i|k} - \bar{r}(k))\|_2^2 + \sum_{j=0}^{N_u-1} \|R\Delta\mathring{r}_{k+j|k}\|_2^2 \tag{4.12a}$$

$$\text{s.t.} \quad x_{c,k|k} = \hat{x}_{c,k|k-1}, \tag{4.12b}$$

$$x_{c,k+i+1|k} = A_f^\eta x_{c,k+i|k} + B_f^\eta \mathring{r}_{k+i|k}, \tag{4.12c}$$

$$y_{c,k+i+1|k} = C_f^\eta x_{c,k+i+1|k} + D_f^\eta \mathring{r}_{k+i|k}, \tag{4.12d}$$

$$\Delta\mathring{r}_{k+N_u+j|k} = 0, \tag{4.12e}$$

$$i = 0, 1, \ldots, N_p - 1, \tag{4.12f}$$

$$j = 0, 1, \ldots, N_p - N_u - 1, \tag{4.12g}$$

where N_p is the prediction horizon, N_u is the control horizon, Q and R denote the weight matrices, $x_{c,k+i|k}$ is the prediction of x at time $k+i$ based on information available at time k, $\Delta\mathring{r}_{k+i|k} = \mathring{r}_{k+i|k} - \mathring{r}_{k+i-1|k}$ is the vector of input increments and $\hat{x}_c$ is the state estimation. In order to cope with the input delay, the open loop optimal control problem is initialized at each time step, with a predicted state estimation, computed by the following Kalman filter:

$$\hat{x}_{c,k+1|k} = (A_f^\eta - LC_f^\eta)\hat{x}_{c,k|k-1} + B_f^\eta - LD_f^\eta \mathring{r}_{k-1} + Ly_k \tag{4.13}$$

where L is the Kalman gain. The Kalman filter guarantees also the mitigation of the noise. Figure 4.4 shows the block scheme of the proposed MPC, where q^{-1} represents the one-step delay operator.

Problem (4.12) can be cast into the parametric unconstrained QP problem

$$\min_z \quad \frac{1}{2} z^T H z + p^T(k) F^T z \tag{4.14}$$

Table 4.1: Hardware and software specifications for considered tests

Parameter	Symbol	Value	Units
DC-DC Buck Converter			
Input Voltage Range		4.75-14	V
Output Voltage Range		0.7-3.6	V
Switching Frequency		400	kHz
Inductance	L	0.9	μH
Inductance Parasitic Resistance	R_l	2.2	mΩ
Capacity	C	470	μF
Transistor Parasitic Resistance	R_{on}	3.6	mΩ
Load Resistance	R	1	Ω
Controllers' Parameters			
Primal Control Frequency	T_s	400	kHz
MPC Control Frequency	T_s^η	100	kHz
Proportional Gain	K_p	0.0195	
Integral Gain	K_i	350	
Prediction horizon	N_p	10	
Control horizon	N_u	5	
Measured output weight	Q	5	
Manipulated variable rate weight	R	0.1	

where $z \in \mathbb{R}^{n_z}$ and $p(k) \in \mathbb{R}^{n_p}$ are the vectors of optimization variables and time-varying affine parameters, respectively. Note that in this formulation the optimization variables correspond to the inputs increments of the PWM duty-cycle that drives the transistors. The parameters' vector for the control of the closed-loop buck converter is $p(k) = [\mathring{r}(k-1)\ \ \hat{x}_c(k)\ \ \bar{r}(k)]'$. The solution z^* of the problem (4.14) is analytic and equal to $z^* = H^{-1}Fp(k)$. Being z^* the optimal sequence of input increments, only the first n_u components are considered and applied to the system. Thus, the solution of the unconstrained MPC problem reduces to a matrix vector product, where the first n_u rows of $H^{-1}F$ are computed offline and stored. However, for such high speed problems as power converters, even the unconstrained solution does not represent a negligible cost for low-power embedded boards. The advantage of the double, multi-rate, loop is the possibility to run MPC in a slower task respect to the primal controller. Considering both the solution of problem (4.14), and the computation of the estimated state (4.13), the complexity c of the control algorithm can be explicitly computed as $c = (2n_p - 1)n_u + (2n_y + 2n_u + 2n_{x_c} - 1)n_{x_c}$.

4.1.2.3 Simulation and experimental results

The proposed control technique was tested both in a simulation environment, and experimentally. The PTD08A010WAD 10A synchronous buck converter

Table 4.2: Performance of VMC and MPC-VMC under unknown load variations

Load	VMC		MPC-VMC	
R[Ω]	τ_r[ms]	τ_s[ms]	τ_r[ms]	τ_s[ms]
0.2	0.51	0.65	**0.31**	**0.39**
0.4	0.61	0.77	**0.35**	**0.45**
0.6	0.67	0.84	**0.38**	**0.49**
0.8	0.70	0.88	**0.40**	**0.51**
1	0.72	0.91	**0.41**	**0.53**
1.2	0.73	0.93	**0.42**	**0.54**
1.4	0.74	0.93	**0.42**	**0.55**
1.6	0.75	0.94	**0.43**	**0.56**
1.8	0.76	0.96	**0.44**	**0.56**
2	0.77	0.97	**0.44**	**0.57**

Table 4.3: Performance comparison of VMC and MPC-VMC with different primal controller's tuning

	VMC		MPC-VMC	
G_m[dB]	τ_r[ms]	τ_s[ms]	τ_r[ms]	τ_s[ms]
5	0.63	1.28	**0.07**	**0.65**
10	0.72	0.91	**0.41**	**0.53**
20	2.00	2.478	**0.73**	**0.95**

Table 4.4: Experimental improvements of MPC-VMC respect to standard VMC under step variations

		VMC	MPC-VMC
Positive Step	τ_r[ms]	0.735	**0.3024**
	τ_s[ms]	0.917	**0.3683**
Negative Step	τ_r[ms]	0.725	**0.3928**
	τ_s[ms]	0.995	**0.4482**

has been used [69] in conjunction with a Delfino F28335 digital power control module, shown in Fig. 4.5, commonly used in power systems, running VMC and MPC-VMC algorithms.

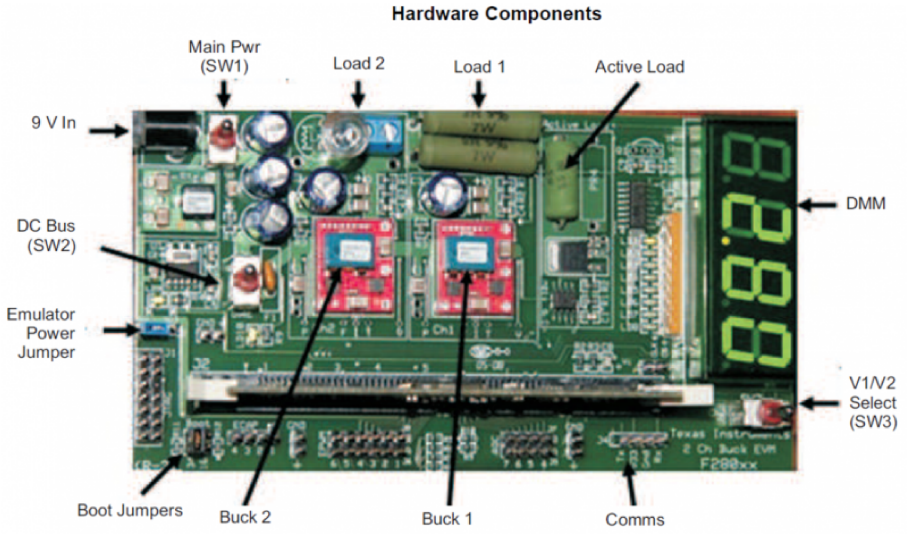

Figure 4.5: Texas Instruments TMDSDCDC2KIT development kit board

They are commercially available, by Texas Instruments©, and the buck converter main parameters are listed in Table 4.14. The control scenario consists into supplying a 1Ω-4W load with 1V and 2V DC voltage. The input supply voltage is 9V, and the switching frequency is set to 400kHz. The simulation tests have been carried out on PSIM® software (by Powersim Inc), which allows for the simulation of both analog and digital components, and control algorithms. The electrical model of PTD08A010WAD, based on [69], is available in PSIM® demo library.

The controller tuning has been reported in Table 4.1. The primal controller, running at 400kHz has been tuned with a standard methodology based on the transfer functions derived from the model (4.1), (4.2). The PI gains have been tuned in order to meet the stability margins as mentioned before, namely a gain margin of 10 dB and a phase margin of more than 45 degrees [68]. The MPC applied to the pre-compensated runs with a sampling frequency of 100kHz, in order to demonstrate that notable control improvements are obtained applied the proposed algorithm even when the reference is changed slower with respect to the primal control frequency.

Two different tests have been proposed. The first consists in a positive step in the reference voltage from 1V to 2V. The second one consists in a negative

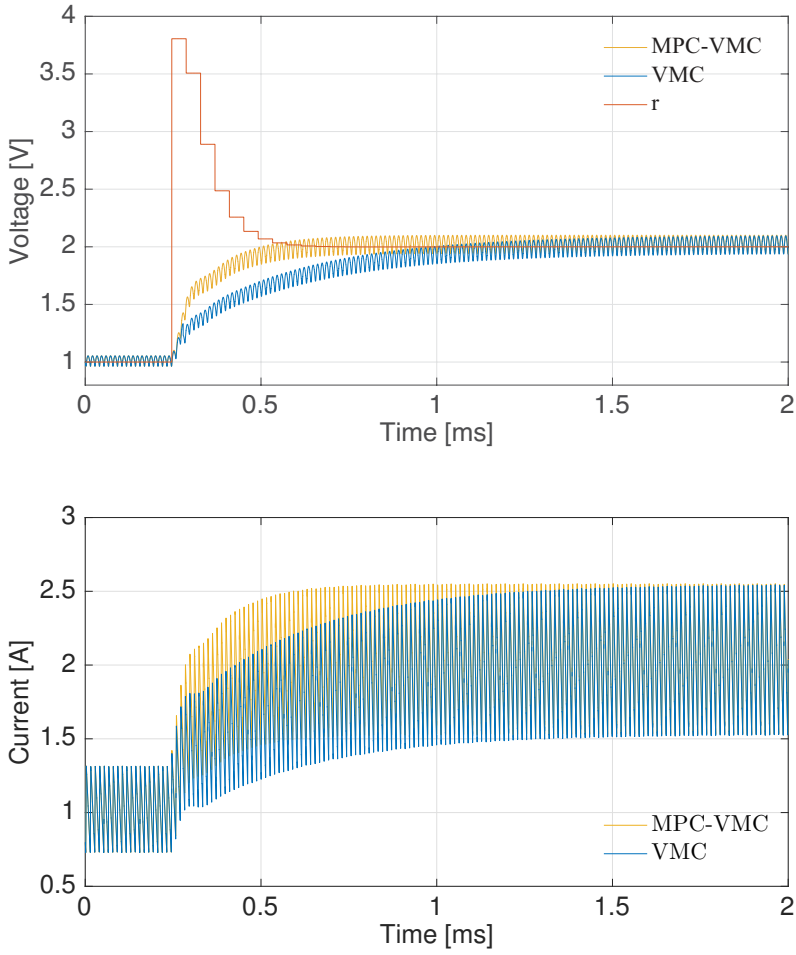

Figure 4.6: Simulation results, with an increasing output voltage step, comparison between standard VMC and and MPC-VMC.

step from 2V to 1V. For both tests the MPC-VMC performance are compared to standard VMC. Obviously, for both VMC and MPC-VMC, the primal controller is the same, namely a PI regulator with the same design parameters. Fig. 4.6 and Fig. 4.7 show the comparison between VMC and MPC-VMC during the transients of a positive and a negative step in the output voltage reference, respectively. Both the inductor current and the output voltage are shown, as well as the dynamically modified reference for the MPC-VMC algorithm. The results demonstrate the improvements in control performance, when applying MPC-VMC, regarding the rise time τ_r and the settling time τ_s. Indeed, during the positive step, MPC-VMC exhibits a reduction of 43.06% and 41.76% of

the rise time and the settling time, respectively. Whereas, during the negative step the rise time and the settling time are reduced by 42.09% and 40.89% respectively.

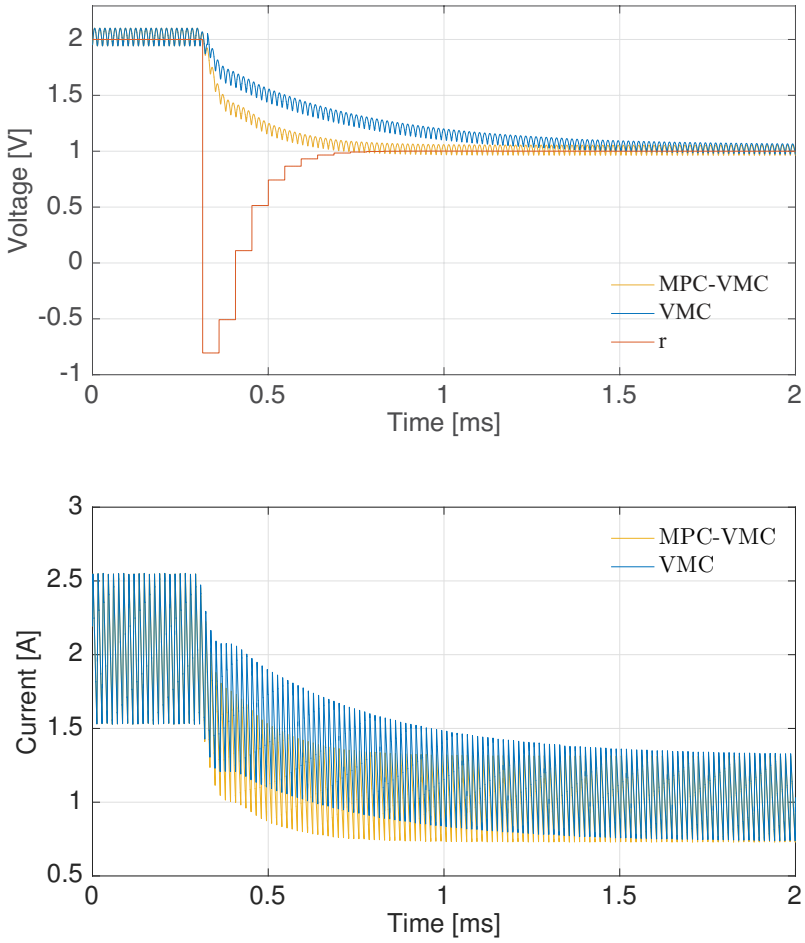

Figure 4.7: Simulation results, with a decreasing output voltage step, comparison between standard VMC and and MPC-VMC.

In order to verify the sensitivity to parameters uncertainties, the two algorithms have been tested under load variations. The tuning of the two controllers is the same of the above experiment, namely Table 4.1, but the load value has been changed from a minimum of 0.2Ω to a maximum of 2Ω. Table 4.2 collects the results when comparing the standard VMC and MPC-VMC in several perturbed scenarios, during a positive voltage reference step. It is evident that MPC-VMC is always able to improve the performance, even when the load R

is significantly changed respect to the value where the controllers have been tuned. The effectiveness of the approach with different primal controller tunings have been also tested, namely a more aggressive primal controller, and a less aggressive one. The purpose of the test is to show that, for different tunings of the primal controller, the MPC-VMC is always able to improve the performance of the standard VMC. Table 4.3 show this comparison for different tuning of the primal controller, changing the gain margin G_m.

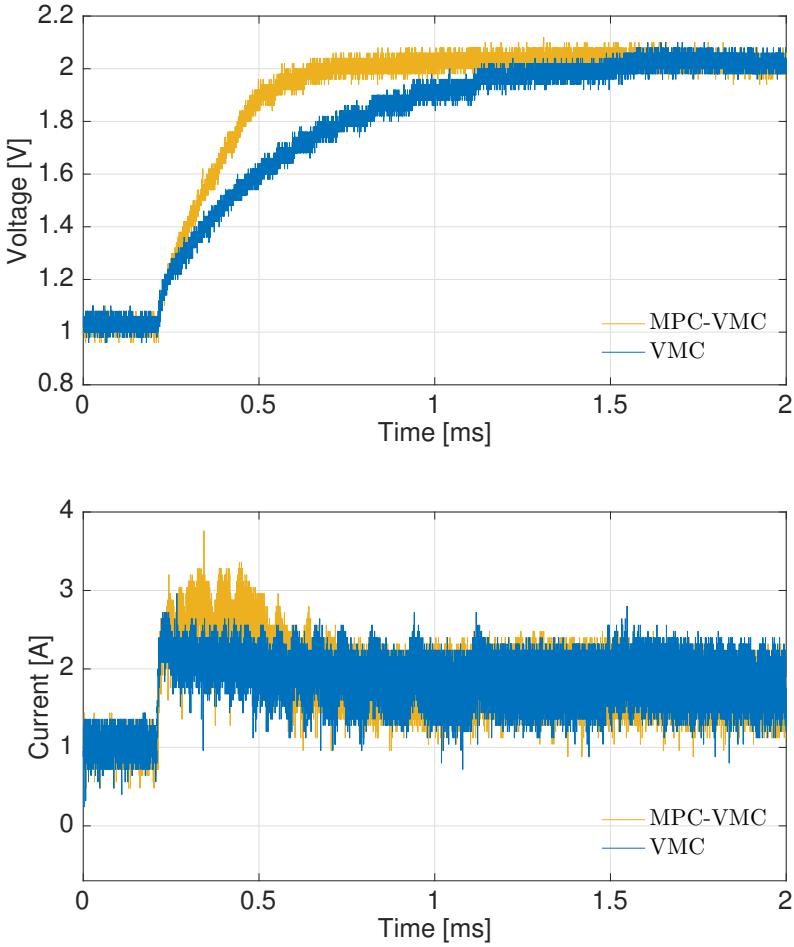

Figure 4.8: Experimental results, with an increasing output voltage step, comparison between standard VMC and and MPC-VMC.

Experimental results on the same DC-DC converter confirm the reliability and feasibility of the approach. Test acquisitions are collected with a Tektronix DPO3014 Digital Phosphor Oscilloscope. The primal controller and the MPC

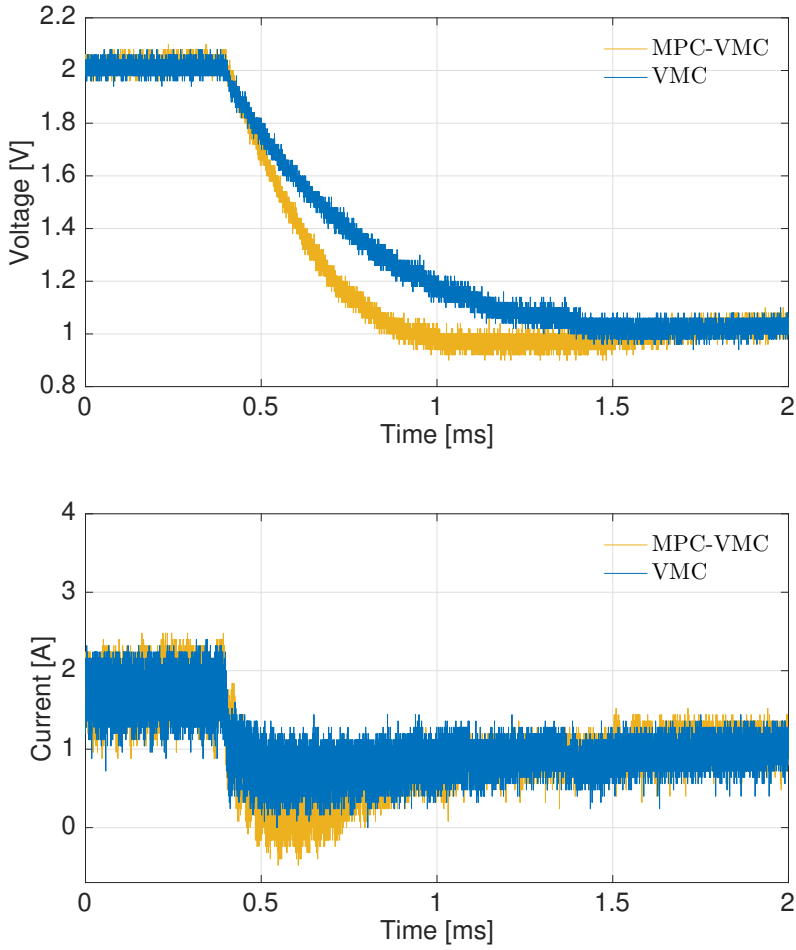

Figure 4.9: Experimental results, with a decreasing output voltage step, comparison between standard VMC and MPC-VMC

have been coded on the F28335 Delfino DSP by Texas Instruments©. The DSP belongs to C2000 series, commonly used in electrical devices control. It runs a 150MHz unit with 32-bits architecture and IEEE-754 single-precision floating-point unit. Two control interrupts regulate the execution of the primal controller and the MPC loop. The first is executed at 400kHz, equal to the switching frequency, whereas the MPC loop runs at 100kHz, following the same setup of simulation test. Fig. 4.8 and Fig. 4.9 present the results for the increasing and decreasing step in the output voltage reference, respectively. The results obtained in the simulation environment have been confirmed by this experimental test. The quantitative comparison of the results is detailed in

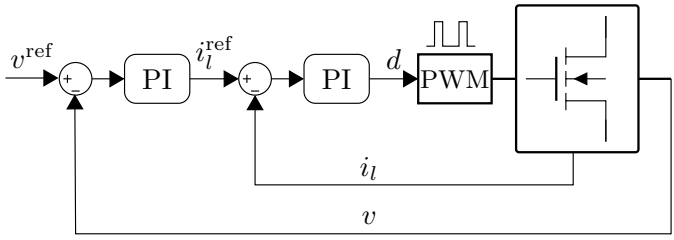

Figure 4.10: Standard Current Mode Control.

Table 4.4. During the positive step signal MPC-VMC guarantees a reduction of 58.86% and 59.84% of the rise time and the settling time, respectively. During the negative step, the rise time and the settling time are reduced by 45.82% and 54.95% respectively. Fig. 4.8 and Fig. 4.9 show also the inductor current which has a faster dynamics in MPC based control, but still keeping the transient peak restrained as expected.

4.1.3 MPC for pre-compensated CMC Buck Converter

Power converter controllers must guarantee output voltage regulation at a desired value. Current Mode Control (CMC) exploits the faster dynamics of the current to provide a responsive control and over current protection, with respect to VMC that presents a single control loop based on voltage feedback and provides slower transient responses performance. For this reason CMC is preferred in high performance applications [66]. The implemented standard CMC block scheme is shown in Fig. 4.10: it is composed by a double control loop with two linear Single-Input Single-Output (SISO) Proportional Integral (PI) regulator [65]. The inner regulator is aided to control the inductor current, reducing the over-shoot, acting directly on the buck convert control input d. The external controller regulates the output voltage to the reference value, providing the reference for the inner current control loop. PIs regulator considered in the realization of the CMC scheme are introduced in the next section.

4.1.3.1 Closed-loop CMC model

The following results will be useful to build the closed-loop state-space system that has to be controlled by the MPC loop. In the following, consider the state space representation of controllers proposed in Definition 4.1.1, the current regulator will be denoted by $\mathcal{S}_{c_1}$ and the voltage controller with $\mathcal{S}_{c_2}$.

Definition 4.1.2. *Consider a linear SISO controller $\mathcal{S}_i$ and a SIMO plant $\mathcal{S}_j$, and let $\mathcal{S}_l \equiv (\mathcal{S}_i, \mathcal{S}_j)$ being the serialized system, such that $u_l \equiv u_i$, $y_l \equiv y_j$*

and $u_j \equiv y_i$. Being $r(k) \in \mathbb{R}$ a reference signal, the closed-loop unity feedback system related to the q-th output $y_{j_{\{q\}}}$ will be denoted by $\mathcal{F}_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)}$, with states x_f and outputs y_f, equal to

$$x_f(k+1) = A_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} x_f(k) + B_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} r(k) \tag{4.15a}$$

$$y_f(k) = C_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} x_f(k) + D_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} r(k) \tag{4.15b}$$

with $x_f \equiv \begin{bmatrix} x_i \\ x_j \end{bmatrix} \in \mathbb{R}^{n_i+n_j}$, $y_{\mathcal{F}} \equiv y_{j_{\{q\}}}$.

Defining for simplicity $d_{\{q\}} \equiv \left[D_{j_{\{q\}}} D_i + I\right]^{-1}$, with $D_{j_{\{q\}}}$ the q-th row of D_j, the following equations hold

$$A_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} = \begin{bmatrix} A_i & 0 \\ B_j B_i & A_j \end{bmatrix} - \begin{bmatrix} B_i \\ B_j D_i \end{bmatrix} \begin{bmatrix} D_{j_{\{q\}}} C_i & C_{j_{\{q\}}} \end{bmatrix} d_{\{q\}} \tag{4.16a}$$

$$B_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} = \begin{bmatrix} B_i \\ B_j D_i \end{bmatrix} - \begin{bmatrix} B_i \\ B_j D_i \end{bmatrix} \begin{bmatrix} D_{j_{\{q\}}} D_i \end{bmatrix} d_{\{q\}} \tag{4.16b}$$

$$C_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} = \begin{bmatrix} D_{j_{\{q\}}} C_i, C_{j_{\{q\}}} \\ D_{j_{\{v\}}} C_i, C_{j_{\{v\}}} - D_{j_{\{v\}}} D_i D_{j_{\{q\}}} C_i, C_{j_{\{q\}}} \end{bmatrix} d_{\{q\}} \tag{4.16c}$$

$$D_{\{q\}}^{(\mathcal{S}_i,\mathcal{S}_j)} = \begin{bmatrix} D_{j_{\{q\}}} D_i \\ D_{j_{\{v\}}} D_i (I - D_{j_{\{q\}}} D_i) \end{bmatrix} d_{\{q\}} \tag{4.16d}$$

where I is the identity matrix of appropriate dimension, and v is the index of the uncontrolled output.

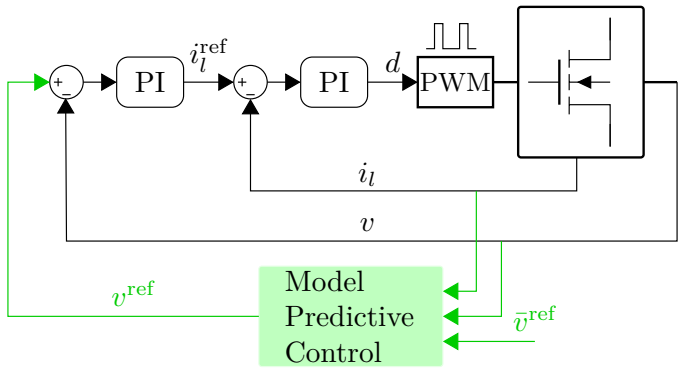

Figure 4.11: Model Predictive Control - Current Mode Control.

Theorem 4.1.3. *Given the synchronous buck converter plant $\mathcal{S}_d$ as in equation (4.2), the two PID controllers $\mathcal{S}_{c_1}$ and $\mathcal{S}_{c_2}$, as in Definition 4.1.1 and the*

feedback constructor in Definition 4.1.2, the complete closed-loop state-space system of a CMC synchronous buck converter, with voltage reference $r(k)$, will be

$$\bar{\mathcal{S}}\left(\bar{A},\bar{B},\bar{C},\bar{D}\right)=\mathcal{F}_{\{2\}}^{\left(\mathcal{S}_{c_2},\mathcal{F}_{\{1\}}^{(\mathcal{S}_{c_1},\mathcal{S}_d)}\right)} \tag{4.17}$$

with extended state and output vectors

$$\bar{x}(k)=\begin{bmatrix}x_{c_2}(k)\\ x_{c_1}(k)\\ x_d((k)\end{bmatrix},\quad \bar{y}(k)=y_d(k). \tag{4.18}$$

Proof. Consider the cascaded CMC structure as in Figure 4.11, the synchronous buck converter SIMO model $\mathcal{S}_d$, and the SISO current controller $\mathcal{S}_{c_1}$ as in Definition 4.1.1. The current feedback will act on the first measurement $y_{d_{\{1\}}}$ and, from Definition 4.1.2, the inner closed-loop system will be again a SIMO system in the form $\mathcal{F}_{\{1\}}^{(\mathcal{S}_{c_1},\mathcal{S}_d)}$, with states x_f, outputs y_f and inputs u_f, such that

$$x_f(k)=\begin{bmatrix}x_{c_1}(k)\\ x_d(k)\end{bmatrix},\ y_f(k)=y_d(k),\ u_f(k)=y_{c_2}(k) \tag{4.19}$$

with $y_{c_2}(k)$ the outer loop's output. Consider now the obtained closed-loop current system as a SIMO system, and the SISO voltage regulator $\mathcal{S}_{c_2}$ applied to $y_{f_{\{2\}}}$, from Definition 4.1.2 such a closed-loop system will assume the form $\mathcal{F}_{\{2\}}^{\left(\mathcal{S}_{c_2},\mathcal{F}_{\{1\}}^{(\mathcal{S}_{c_1},\mathcal{S}_d)}\right)}$ as in (4.17), which will be referred to as $\bar{\mathcal{S}}\left(\bar{A},\bar{B},\bar{C},\bar{D}\right)$, that is the prediction model for the MPC design. Furthermore, Equation (4.19) proves that (4.18) holds, as

$$\bar{x}(k)=\begin{bmatrix}x_{c_2}(k)\\ x_f(k)\end{bmatrix}=\begin{bmatrix}x_{c_2}(k)\\ x_{c_1}(k)\\ x_d(k)\end{bmatrix},\ \bar{y}(k)=y_f(k)=y_d(k) \tag{4.20}$$

and $\bar{u}(k)\equiv v^{\text{ref}}(k)$ which is the voltage reference that will be modified by the MPC controller. □

The linear system $\bar{\mathcal{S}}(\bar{A},\bar{B},\bar{C},\bar{D})$ represents the cascaded CMC closed-loop of the synchronous buck converter, with two PID regulators. However in DC-DC converters simple PI controllers are usually employed [65], and this holds true also for the simulation results presented.

4.1.3.2 MPC-CMC

The MPC loop is designed with the Linear Time Invariant (LTI) prediction model $\bar{\mathcal{S}}(\bar{A}, \bar{B}, \bar{C}, \bar{D})$. The MPC optimization problem is formulated as in the follows:

$$\begin{aligned}
\min_{\Delta u} \quad & \sum_{i=1}^{N_p-1} \|Q(\bar{y}_{k+i|k} - \bar{v}^{\text{ref}}(k))\|_2^2 + \sum_{j=0}^{N_u-1} \|R\Delta v^{\text{ref}}_{k+j|k}\|_2^2 + \\
& + \|P(\bar{y}_{k+N|k} - \bar{v}^{\text{ref}}(k))\|_2^2 && \text{(4.21a)} \\
\text{s.t.} \quad & \bar{x}_{k|k} = x_d(k), && \text{(4.21b)} \\
& \bar{x}_{k+i+1|k} = \bar{A}\bar{x}_{k+i|k} + \bar{B}v^{\text{ref}}_{k+i|k}, && \text{(4.21c)} \\
& \bar{y}_{k+i+1|k} = \bar{C}\bar{x}_{k+i+1|k}, && \text{(4.21d)} \\
& \Delta v^{\text{ref}}_{k+N_u+j|k} = 0, && \text{(4.21e)} \\
& i = 0, 1, \ldots, N_p - 1, && \text{(4.21f)} \\
& j = 0, 1, \ldots, N - N_u - 1, && \text{(4.21g)}
\end{aligned}$$

with N_p and N_u the prediction and the control horizons, respectively, Q, R, and P denote the weight matrices, $x_{k+i|k}$ is the prediction of x at time $k+i$ based on information available at time k, and $\Delta v^{\text{ref}}_{k+i|k} = v^{\text{ref}}_{k+i|k} - v^{\text{ref}}_{k+i-1|k}$ is the vector of input increments. According to Figure 4.11, v^{ref} is the time varying reference that will be applied to the primal controller, and $\bar{v}^{\text{ref}}$ is the actual voltage value, usually constant, that the entire controller algorithm has to track. Substituting the equality constraints into the cost function, the problem (4.21) is casted into a parametric QP, easier to handle,

$$\min_z \quad q(z) \triangleq \frac{1}{2} z' H z + p^T(k) F' z \qquad \text{(4.22a)}$$

where z denotes the optimization variables and $p(k)$ are the time-varying affine parameters that change at each sampling time. This parameters' vector is $p_k = [v^{\text{ref}}(k-1)\ \bar{x}(k)\ \bar{v}^{\text{ref}}]'$. The optimal solution z^* is obtained at each sampling time, and the first n_u components are applied to the system. The solution is analytical in this case, and equal to $z^* = -H^{-1}Fp(k)$.

4.1.3.3 Simulation and experimental results

Both the electrical circuit and the digital controller have been coded in PSIM® software, by Powersim Inc. The synchronous DC-DC buck converter was modeled following the specification of the commercially available PTD08A010WAD, a 10A synchronous buck whose parameters are detailed in Table 4.5.

The proposed MPC-CMC was tested with respect to standard CMC in simulation and experimental tests, considering the setup already proposed in

Table 4.5: Hardware and software specifications for simulation and experimental tests

Parameter	Symbol	Value	Units
DC-DC Buck Converter			
Input Voltage Range		4.75-14	V
Output Voltage Range		0.7-3.6	V
Switching Frequency		400	kHz
Inductance	L	0.9	μH
Inductance Parasitic Resistance	R_l	2.2	mΩ
Capacity	C	470	μF
Transistor Parasitic Resistance	R_{on}	3.6	mΩ
Load Resistance	R	1	Ω
Controllers' Parameters			
Primal Control Frequency	T_s	400	kHz
MPC Control Frequency	T_s^η	100	kHz
Current Proportional Gain	K_p^I	0.019	
Current Integral Gain	K_i^I	375	
Voltage Proportional Gain	K_p^V	3.75	
Voltage Integral Gain	K_i^V	5800	
Prediction horizon	N_p	10	
Control horizon	N_u	3	
Measured output weight	Q	1	
Manipulated variable rate weight	R	5	

Table 4.6: Performance comparison of CMC and MPC-CMC under unknown load variations for a positive reference step

Load	CMC		MPC-CMC	
R[Ω]	τ_r[ms]	τ_s[ms]	τ_r[ms]	τ_s[ms]
0.2	3.410	5.234	**3.087**	**5.115**
0.4	2.351	4.709	**2.041**	**4.570**
0.6	1.687	2.311	**1.577**	**2.028**
0.8	1.005	1.232	**0.993**	**1.094**
1	0.107	0.420	**0.075**	**0.320**
1.2	0.093	0.384	**0.071**	**0.301**
1.4	0.089	0.351	**0.069**	**0.276**
1.6	0.078	0.299	**0.066**	**0.255**
1.8	0.069	0.232	**0.061**	**0.201**
2	0.057	0.121	**0.055**	**0.118**

Table 4.7: Simulation results of MPC-CMC respect to standard CMC under step variations

		τ_r [ms]	τ_s [ms]	i_l^{os} [A]
Positive Step	CMC	0.1073	0.4205	5.683
	MPC-CMC	**0.0752**	**0.3202**	**4.338**
Negative Step	CMC	0.1088	0.4221	-8.681
	MPC-CMC	**0.0655**	**0.3362**	**-5.554**

Table 4.8: Experimental results of MPC-CMC respect to standard CMC under step variations

		τ_r [ms]	τ_s [ms]	i_l^{os} [A]
Positive Step	CMC	0.1436	0.3736	9.531
	MPC-CMC	**0.0863**	**0.3276**	**4.714**
Negative Step	CMC	0.1994	0.4402	-6.912
	MPC-CMC	**0.0852**	**0.3744**	**-3.535**

Sec. 4.1.2.3. Two different step tests have been considered. The first consists in a positive step in the reference voltage from 1V to 2V. The second one consists in a negative step from 2V to 1V. For both tests the MPC-CMC performance are compared to standard CMC. For both compared controllers the primal control loop is the same, with two PI regulators with the same design parameters. Fig. 4.12 and Fig. 4.13 show the comparison between CMC and MPC-CMC during the transients of a positive and a negative step in the output voltage reference, respectively. Both the inductor current and the output voltage are shown, as well as the original and the dynamically modified reference for the MPC-CMC algorithm. Simulated step test results are grouped in Tab.4.7, where CMC and MPC-CMC are compared considering the rise time τ_r, the settling time τ_s and the inductor current peak value i_l^{os}. Both results related to positive and negative steps demonstrate the improvements in control performance when applying the MPC-CMC. In positive step reference test the MPC-CMC allows to reduce the rise time of 29.91%, the settling time of 23.85 and the inductor current peak of 23.66%. Negative step test simulation results confirm MPC-CMC improvement, showing a reduction of 39.79%, 20.35%, 36.02% of rise and settling time and current peak, respectively.

In order to verify the sensitivity to parameters uncertainties, both algorithms have been compared in simulation to load variation tests. The tuning of the two controllers is the same of the previous test, summarized Table 4.14, but the load

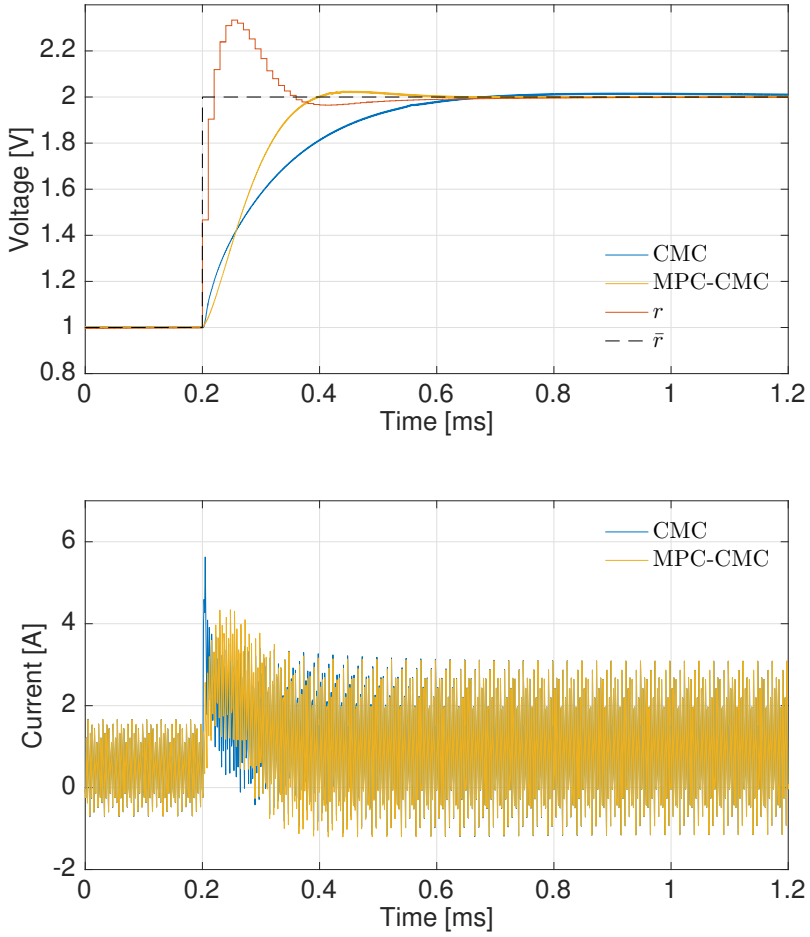

Figure 4.12: Simulation results, with an increasing output voltage step, comparison between standard CMC and and MPC-CMC.

value has been changed from a minimum of 0.2Ω to a maximum of 2Ω, from the 10% to the 200% of the nominal value. Table 4.6 groups the results when comparing CMC and MPC-CMC algorithms in different perturbed scenarios, during a positive voltage reference step. It is evident that MPC-CMC is always able to improve the performance, even when the load R is significantly changed respect to the value where the controllers have been tuned.

Experimental results on the same DC-DC converter confirm the reliability and feasibility of the approach. Two control interrupts regulate the execution of the primal controller and the MPC loop. The first is executed at 400kHz, equal to the switching frequency, whereas the MPC loop runs at 100kHz, following

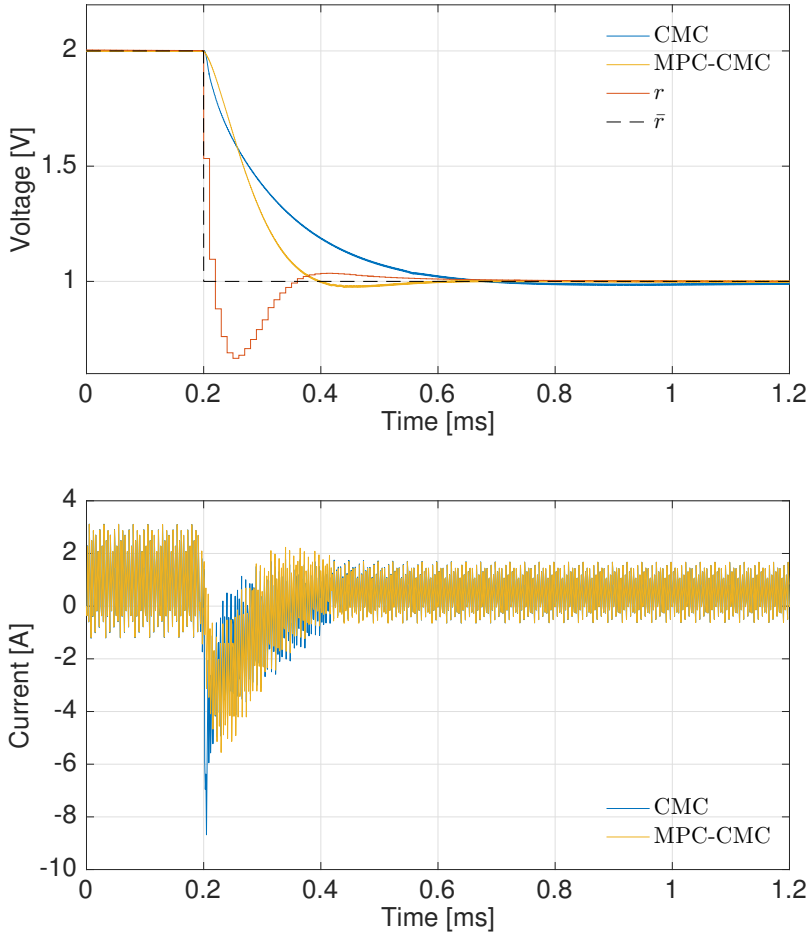

Figure 4.13: Simulation results, with a decreasing output voltage step, comparison between standard CMC and MPC-CMC

the same setup of simulation test. Fig. 4.14 and Fig. 4.15 present the results for the increasing and decreasing step in the output voltage reference, respectively. The results obtained in the simulation environment have been confirmed by this experimental test. The quantitative comparison of the results is detailed in Table 4.8. During the positive step signal MPC-CMC guarantees a reduction of 41.78%, 12.31% and 48.17% of rise time, settling time and current peak, respectively. During the negative step, the rise time, the settling time are reduced of the 57.27%, 14.95% and the current peak decreases of the 48.85%.

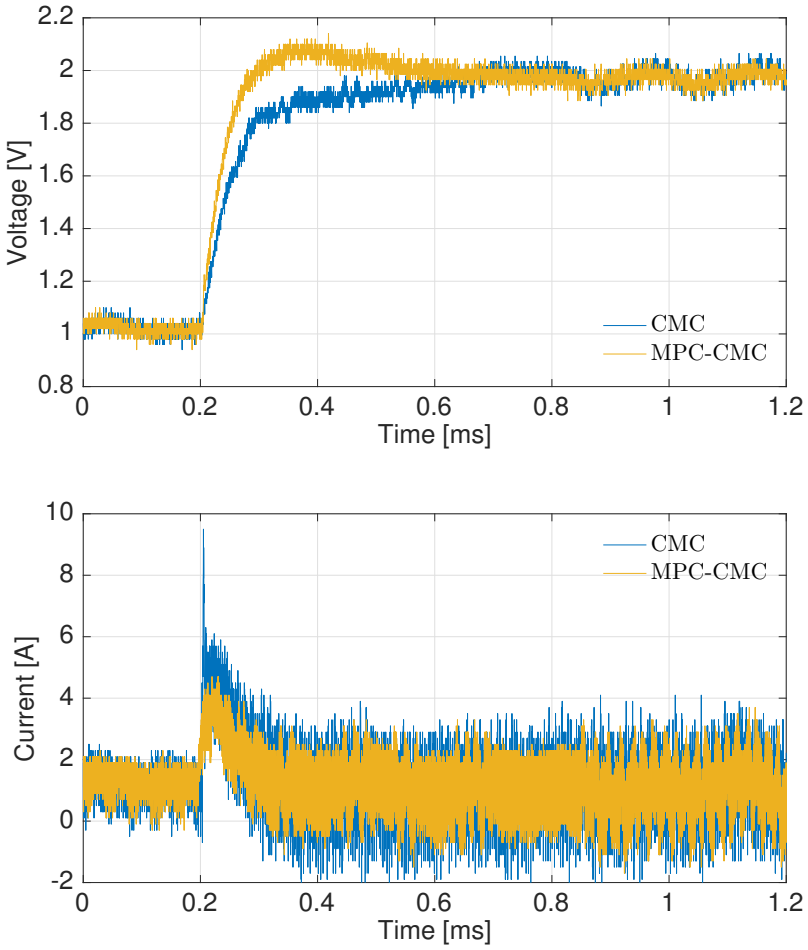

Figure 4.14: Experimental results, with an increasing output voltage step, comparison between standard CMC and and MPC-CMC.

4.2 Piezoelectric actuators

Piezoelectric actuators (PAs) systems have been widely applied in several fields requiring high-accuracy movement [70], such as micromanipulation systems [71] and scanning probe microscopy [72]. They are featured by high-frequency, fast dynamics, string stiffness and ultra-fine resolution [73]. Moreover PAs exhibit several critical problems such as disturbances and positioning errors related to hysteresis [74], drift of the output positioning [75], vibration phenomenons related to the dynamics behavior of the electromechanics system [76]. In order to overcome these drawbacks, several models and control approaches have been proposed [77] enabling PAs' nominal positioning accuracy and displace-

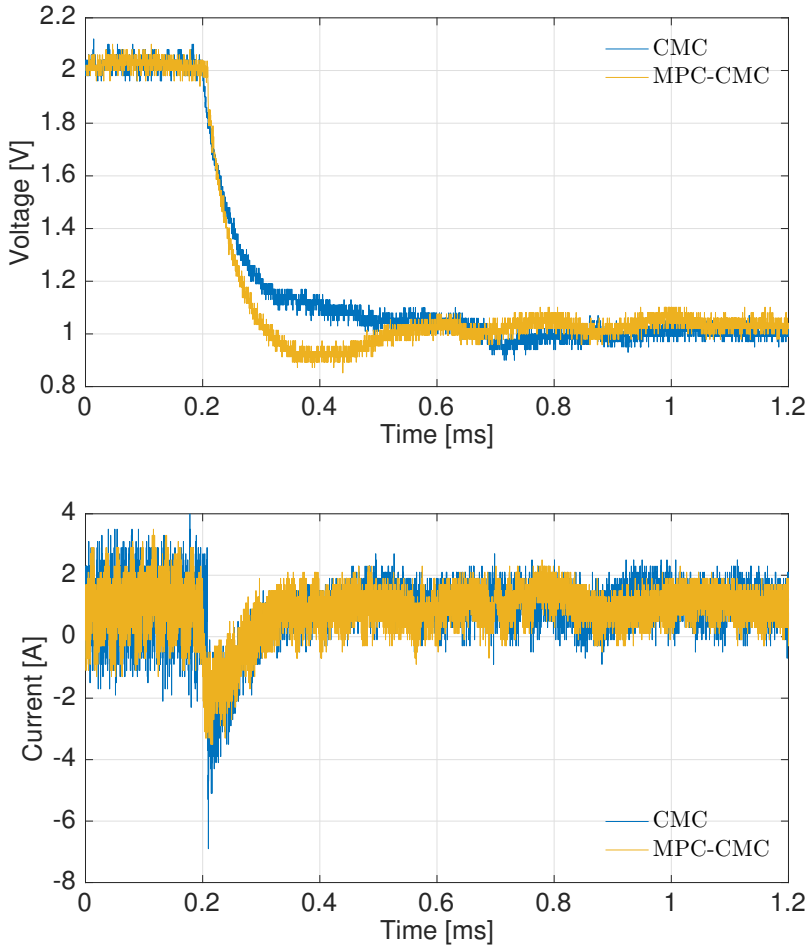

Figure 4.15: Experimental results, with a decreasing output voltage step, comparison between standard CMC and MPC-CMC

ment features. In particular, several dynamical models have been proposed to describe and reduce hysteresis (differential equation and operator-based models [78], [79]) and creep (linear and non-linear models [80]) effects, while identified linear models are usually employed to describe vibrations noise. Considering plant and noise models, different advanced control policies have been suggested in literature, with particular attention to robust control policies, such as Sliding Mode Control [81], H_∞ control [82] and adaptive control [83]. Anyway, the standard for commercial PAs systems is still the Proportional-Integral (PI) control, allowing to guarantee prescribed stability margins by a simple tuning, also in presence of noise and disturbance [84]. Alternative policies to standard

PI controllers is here proposed, by the introduction of Model Predictive Control (MPC) policy. MPC is becoming increasingly interesting in piezo-driven micro-positioning system field [85], also combined with robust control policies [86]. Furthermore, several solutions have been proposed in MPC control law formulation to directly reduce undesired behaviors introduced by PAs nonlinearities [87]. MPC has an implicitly built-in robustness property [88] but requires *i*) an accurate plant model, *ii*) a hard computational effort to provide iteratively the optimal control signal. In the reported activities the required high fidelity plant model is obtained by an identification procedure: forcing the open-loop plant by a set of appropriate input signals, the recorded output are used by a least square identification algorithm to compute the needed model. System identification is a standard solution to model piezoelectric systems due to the linear input-output behavior of real systems [89]. The result of the identification is a discrete Multi-Input Multi-Output (MIMO) Linear Time-Invariant (LTI) model used to design the proposed controller. As mentioned, also computational burden is an important issue in MPC and in this thesis unconstrained solutions have been applied to over limits related to the introduction of MPC on commercial hardware, reducing the time elapsed to compute the control law solving. In Sec. 4.2.1 and 4.2.2 the proposed MPC and RG control strategies have been reported, respectively, showing the effectiveness of the introduced controllers by simulations and experimental verifications.

4.2.1 MPC for multiaxis piezoactuators

4.2.1.1 Expertimental Setup

Tests have been performed by a NI cRIO-9004® acquisition system connected to the piezoelectric actuators (PAs) system. cRIO® is an automation control system using a FPGA low level control module, a real-time control module, an internal processor running to a maximum of 40 MHz and two analog input/output modules characterized each one by two 16 bit Analog to Digital (ADC) and Digital to Analog (DAC) converter. The proposed controllers are implemented with a sample rate of 2 kHz. cRIO® is connected to the strain gauge sensors of the Piezostage (PI P-611.3S®) and to a control amplifier (PI E-505 Amplifier Module®). The amplifier allows to generate an adequate control voltage for the piezostage, amplifying cRIO® control signal from a range of 0-10V to the required piezo input range of 0-100V. The triaxial driver controls the position of the Strain Gauge Sensor (SGS) with a resolution of 0.2 nm in open loop travel. SGS are directly connected to the analog input module of the cRIO®, allowing to acquire stage movement. The experimental setup is presented in Fig. 4.16.

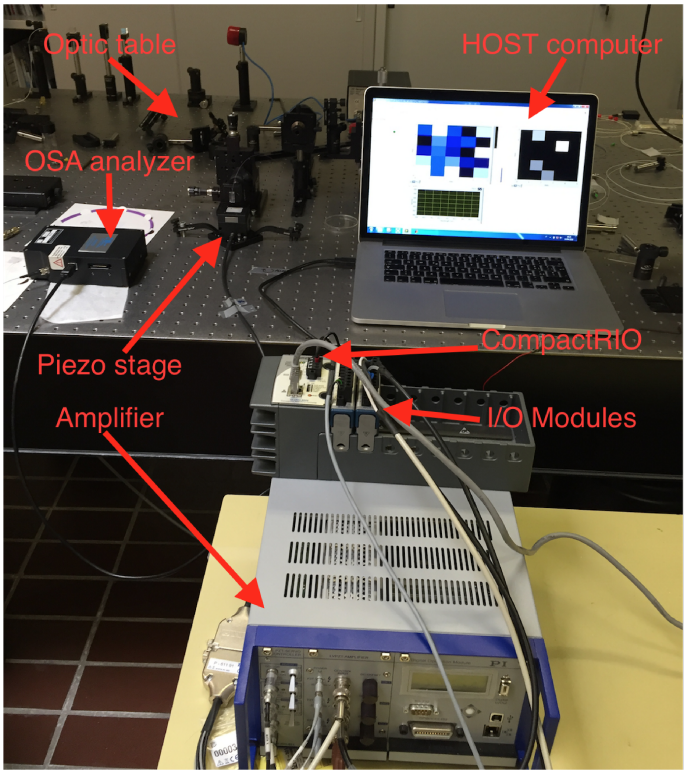

Figure 4.16: Multiaxis piezoactuators experimental setup

4.2.1.2 Identification of the multiaxis piezoelectric system

In the identification procedure, the multi-axis piezo input voltages referred to X, Y and Z axis are named u_1, u_2 and u_3 respectively. In the same way the measured output displacements are y_1 (X axis), y_2 (Y axis) and y_3 (Z axis). Signals data are acquired with a sampling rate of 2 kHz. The system is identified forcing the open-loop real plant with several sets of input steps and recording related output. Input signals have time width bigger than the settling time of the open-loop system, considered of 50 ms. The smaller input rate causing a detectable output variations is $|\Delta u_i| \geq 0.01$ V and input signals bounds are $u_i \in [-1, 10]$ V with $i = 1, 2, 3$. During data acquisition the driver of each axis has been forced, in order to model any possible coupling effects between different axis. Collected input and output data are used to identify a LTI plant model by a least square identification algorithm. The identified model is tested comparing computed outputs with respect to different sets of acquired data not considered in the identification procedure. Consider the difference between computed and acquired outputs, validation tests results are

evaluated in terms of the Normalized Root Mean Square Error (NRMSE). The goodness of the identified model fit is indicated by a NRMSE average value of 89%. In Fig. 4.17 and 4.18 validation tests results are reported. Output data

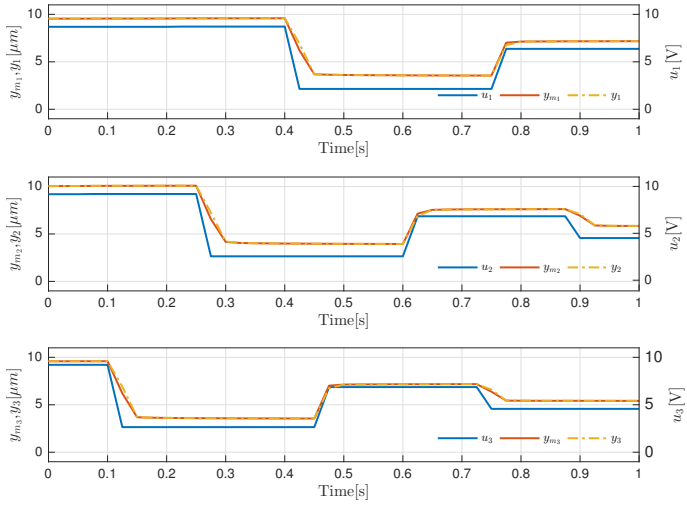

Figure 4.17: Identification test results, with random generated input signals.

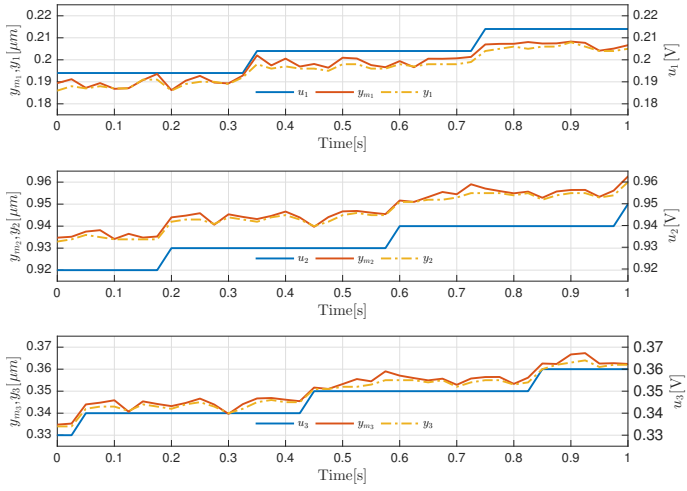

Figure 4.18: Identification test results, with small amplitude input signals.

computed by the identified model are indicated by y_{m_1}, y_{m_2} and y_{m_3} for X, Y, and Z axis respectively. In both figures the identified model allows to fit with good precision the measured outputs. Fig. 4.17 shown the fit between real plant and identified model forced by random amplitude input steps. Fig. 4.18 groups results of comparison forcing the controlled plant with small amplitude

input step signals. Both results illustrate the validity of the identified model. The discrete identified model in state-space form is given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{4.23}$$

where $u(k) = \begin{bmatrix} u_1(k) & u_2(k) & u_3(k) \end{bmatrix}'$ is the input vector of dimension $n_u = 3$, $y(k) = \begin{bmatrix} y_1(k) & y_2(k) & y_3(k) \end{bmatrix}'$ is the output vector with dimension $n_y = 3$, the number of state variables is $n_x = 6$ and the sample time is $T_s = 0.0005$ s. This model will be used in the following to design the control law.

4.2.1.3 MPC design

In the proposed approach the optimization problem considers a quadratic cost function subject to the linear equality constraints representing the model

$$\min_{\Delta u} \quad \sum_{i=1}^{N_p} \|W_y(y_{k+i|k} - r_k)\|_2^2 + \sum_{j=0}^{N_u-1} \|W_{\Delta u}\Delta u_{k+j|k}\|_2^2 \tag{4.24a}$$

$$\text{s.t.} \quad x_{k|k} = x(k), \tag{4.24b}$$

$$x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i|k}, \tag{4.24c}$$

$$y_{k+i+1|k} = Cx_{k+i+1|k}, \tag{4.24d}$$

$$\Delta u_{k+N_u+j|k} = 0, \tag{4.24e}$$

$$i = 0, 1, \ldots, N_p - 1, \tag{4.24f}$$

$$j = 0, 1, \ldots, N_u - 1 \tag{4.24g}$$

where N_p is the prediction horizon, N_u is the control horizon, $W_{\Delta u}$ and W_y are the weights matrices, with $W_{\Delta u}$ invertible, $x_{k+i|k}$ denotes the prediction of the variable x at time $k+i$ based on the information available at time k, $\Delta u_{k+i|k}$ is the vector of the input increments, with $u_{k-1|k} = u(k-1)$, r_k is the output reference. Problem (4.24) can be casted into the parametric Quadratic Programming (QP) problem

$$\min_{z} \quad \frac{1}{2} z'Hz + \rho_k' F' z \tag{4.25}$$

where $\rho_k \in \mathbb{P}$ is the vector of MPC parameters of dimension n_ρ, with $\mathbb{P} \subset \mathbb{R}^{n_\rho}$ a bounded set of interest, $z \in \mathbb{R}^{n_z}$ is the vector of optimization variables, $H \in \mathbb{R}^{n_z \times n_z}$ is a symmetric and positive definite matrix, $F \in \mathbb{R}^{n_z \times n_\rho}$. The parameters' vector for the control of the closed-loop system is $p(k) =$

Table 4.9: Controllers' specifications for simulation and experimental tests

Parameter	Value
Sample time T_s	0.0005 s
PI Parameters	
K_p X axis	0.8
K_i X axis	750
K_p Y axis	0.8
K_i Y axis	750
K_p Z axis	0.8
K_i Z axis	750
MPC Parameters	
Prediction horizon	150
Control horizon	50
Measured variables weights	$\begin{bmatrix} 6.3 & 6.3 & 6.3 \end{bmatrix}$
Manipulated variables rate weights	$\begin{bmatrix} 0.88 & 0 & 0 \\ 0 & 0.88 & 0 \\ 0 & 0 & 0.88 \end{bmatrix}$

$\begin{bmatrix} u(k-1) & x(k) & r(k) \end{bmatrix}'$. The solution z^* of the problem (4.25) is analytic and equal to $z^* = H^{-1}Fp(k)$. Being z^* the optimal sequence of input increments, only the first n_u components are considered and applied to the system. In the presented solution a standard Kalman filter is considered and, given the state estimation computational effort c_K as in [90], the complexity c of the MPC control law can be explicitly computed as $c = (2n_p - 1)n_u + c_K$.

4.2.1.4 Simulation and Expertimental Results

Both simulation and experimental results have been considered to test the proposed control law. Simulations are realized in the Mathworks© simulation environment testing the controller through the identified model. Test scenario consists into drive simultaneously controlled system outputs with step signals of amplitude 0.2 μm. MPC performances are compared with respect to results obtained by PI controllers. PI parameters are tuned guaranteeing prescribed stability margins, namely a gain margin of 5 dB and a phase margin of 50 degrees [84]. Both MPC and PI control tuning parameters, both obtained by a trial and error approach, are reported in Tab. 4.9. Figures 4.19 and 4.20 present simulation results comparison. Fig. 4.19 considers simulated controlled output behavior, showing the slower PI control results with respect to MPC performances. Simulation results are also summarized and presented in Tab. 4.10 with respect to the considered performance indices: rise time τ_r, settling time τ_s and the Integral Square Error (ISE). MPC policy improves

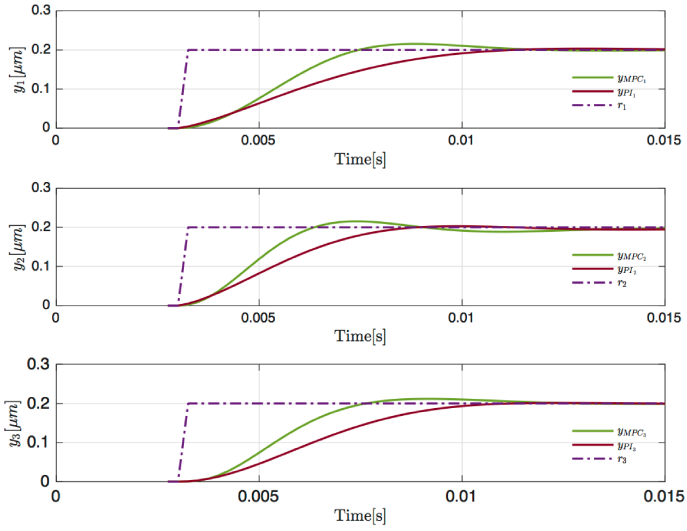

Figure 4.19: Simulation results, comparison between PI and MPC output results.

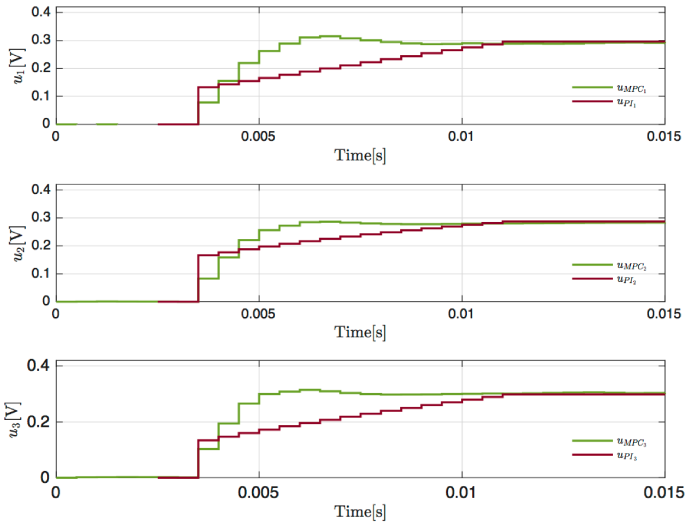

Figure 4.20: Simulation results, comparison between PI and MPC control efforts.

control results, and the faster response is related to a reduced error for each controlled axis, with an average rise time and settling improvement of 20% and 13% respectively and an average ISE decrease of 5%. In Fig. 4.20 the simulated control efforts have been presented. Due to the prediction, MPC control signals present an aggressive behavior to reduce the rise time and also to limit

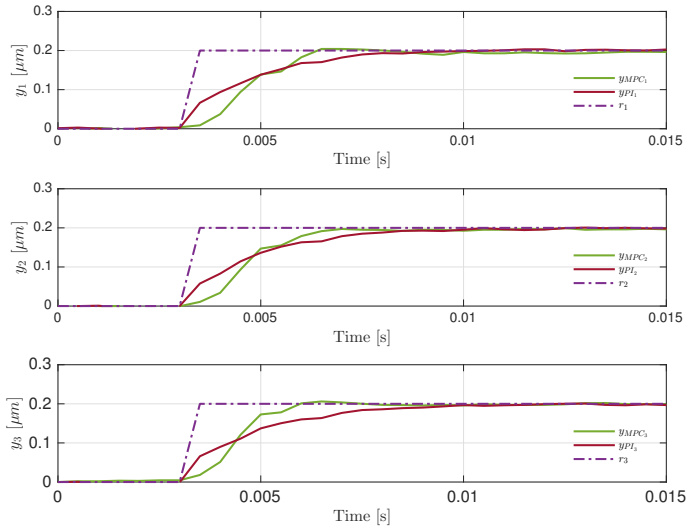

Figure 4.21: Experimental results, comparison between PI and MPC output results.

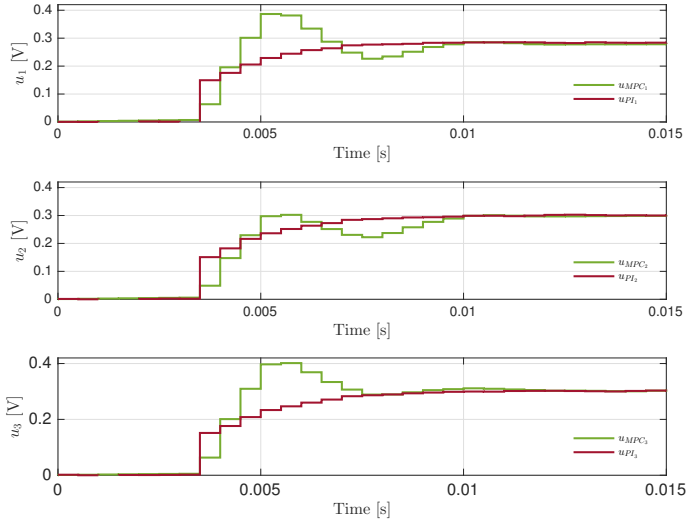

Figure 4.22: Experimental results, comparison between PI and MPC control efforts.

the output signals overshoot. Experimental results on the real PAs system confirm the performance of simulated controllers. Table 4.11 groups data of experimental step input tests. MPC allows to reduce the rise time of the 21% with respect to PI elapsed time. Also the settling time is improved by the MPC with respect to the PI, reducing the elapsed time of the 25%. The average error

Table 4.10: Simulation results: comparison of PI and MPC under step variations

		PI	MPC
X axis	τ_r[ms]	0.006	**0.004**
	τ_s[ms]	0.009	**0.008**
	ISE	0.5213	**0.4823**
Y axis	τ_r[ms]	0.005	**0.004**
	τ_s[ms]	0.007	**0.006**
	ISE	0.5065	**0.4796**
Z axis	τ_r[ms]	0.006	**0.005**
	τ_s[ms]	0.010	**0.009**
	ISE	0.4917	**0.4632**

computed by the ISE during the transient shows an average reduction of the 6%. Fig. 4.21 presents the controlled real plant outputs comparison. Such as in the simulation tests, the MCP controlled output is driven faster to the reference value with respect to the PIs' signal, avoiding overshoots and oscillations. In Fig. 4.22 control efforts are reported. MPC control inputs present an initial aggressive behavior then reduced to prevent the output overshoots by the prediction, similarly to the simulated test results of Fig. 4.19 and 4.20. Both numerical and graphical results validate the proposed approach by the consistency of percentage improvement between the experimental and simulated tests.

4.2.2 MPC for pre-compensated multiaxis piezoactuators

4.2.2.1 Expertimental Setup

Figure 4.23 shows the instrumentation used in the experimental setup. Tests have been performed by the MicroDAQ E2000 acquisition system by Microdaq®. The MicroDAQ provides a Texas Instruments© C6000 DSP core, an ARM9 core and a set of I/O analog ports featured by a maximum sampling frequency of 600 ksps and a resolution of 16 bit. The PA system is composed of a Piezostage (PI P-611.3®) with movement range of $100 \times 100 \times 100$ μm and 1 nm of resolution. The driver moves a Strain Gauge Sensor (SGS) characterized by a resolution on 0.2 nm. The MicroDAQ module is connected to a control amplifier (PI E-505 Amplifier Module®) allowing to generate an adequate control voltage for the PA system amplifying MicroDAQ output control signal from $[-1, 10]$ V to the piezostage input signals range of $[-10, 100]$ V. The amplifier provides an

Table 4.11: Experimental results: comparison of PI and MPC under step variation

		PI	MPC
X axis	τ_r[ms]	0.005	**0.004**
	τ_s[ms]	0.007	**0.005**
	ISE	0.5089	**0.4660**
Y axis	τ_r[ms]	0.005	**0.004**
	τ_s[ms]	0.007	**0.005**
	ISE	0.4837	**0.4524**
Z axis	τ_r[ms]	0.005	**0.004**
	τ_s[ms]	0.007	**0.005**
	ISE	0.4713	**0.4570**

internal set of analog Proportional-Integral (PI) controllers representing the low-level controllers of the PA system.

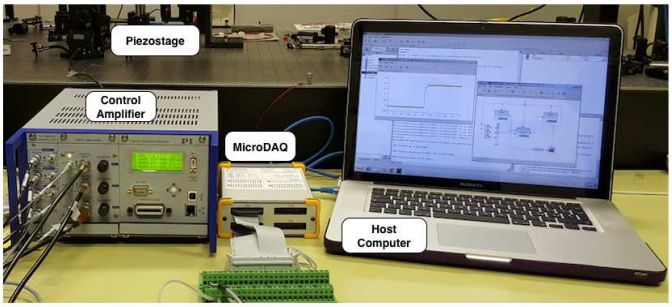

Figure 4.23: Experimental Setup

4.2.2.2 Identification of the pre-compensated system

For the identification procedure, input references for the PI pre-compensated plant, referred to the X, Y and Z axis drivers, are named r_1, r_2 and r_3, respectively. The measured output displacements are y_1 (X axis), y_2 (Y axis) and y_3 (Z axis). Signals data are acquired with a sampling rate of 20 kHz. The identification is realized forcing the closed-loop real plant, driven by the analog PI controller, by several sets of input steps and recording the related outputs. Reference signals have time duration bigger than the settling time of the open-loop system, considered of 120 ms. The smaller reference rate causing a detectable output variations is $|\Delta u_i| \geq 0.05$ V and the input signals

bounds are $u_i \in [-1, 10]$ V with $i = 1, 2, 3$. During the data acquisition the axis drivers were forced together in order to model any possible coupling effects between the axes. Collected input and output data are used to identify a LTI plant model by a least square identification algorithm. The identified model is tested comparing the computed outputs of the LTI plant model with respect to different sets of acquired data not considered in the identification procedure. Considering the difference between the computed and the acquired outputs, validation tests results are evaluated in terms of the Normalized Root Mean Square Error (NRMSE). The goodness of the identified model is proven by a NRMSE average value of 95%. In Fig. 4.24 and Fig. 4.25 validation results

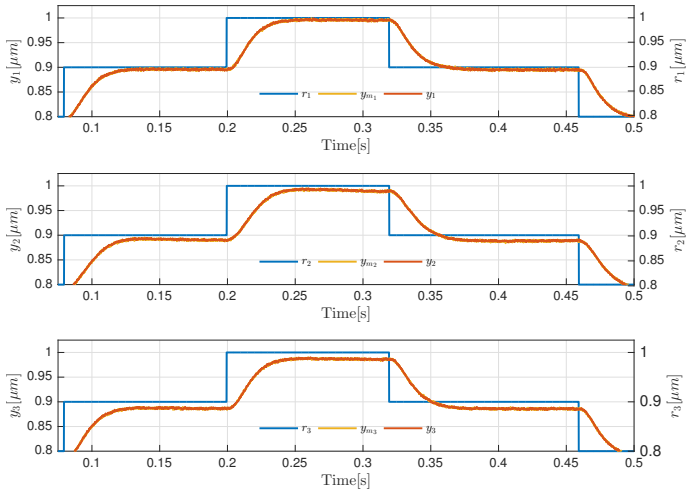

Figure 4.24: Identification test results, with step input signals.

have been presented. Output data computed by the identified LTI model are indicated in Fig. 4.24 by y_{m_1}, y_{m_2} and y_{m_3} for X, Y, and Z axis, respectively. Results in figures show how the identified model fits with good precision the measured outputs. In particular Fig.4.24 shows the fit between the real plant and the identified model forced by steps of fixed amplitude. Fig.4.25 shows the errors (e_x, e_y and e_z) between the outputs of the identified LTI model and the outputs of the real plant. The reduce amplitude of the errors (average value of 0 μm and variance of 0.002 μm, equal to the 1% of the amplitude of the signal variation) proves the goodness of the model.

The proposed MPC controllers have been tuned on the identified discrete LTI model:

$$\begin{aligned} x(k+1) &= Ax(k) + Br(k) \\ y(k) &= Cx(k) \end{aligned} \tag{4.26}$$

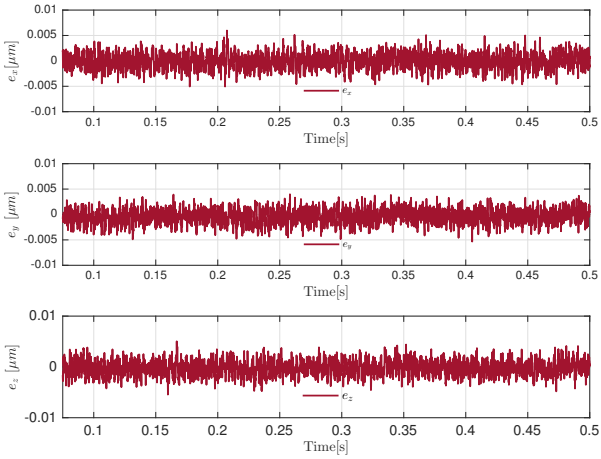

Figure 4.25: Identification test results, with step input signals.

where A, B, C are the identified matrices , $r(k) = \begin{bmatrix} r_1(k) & r_2(k) & r_3(k) \end{bmatrix}'$ is the reference vector for the PI pre-compensated system of dimension $n_r = 3$, $y(k) = \begin{bmatrix} y_1(k) & y_2(k) & y_3(k) \end{bmatrix}'$ is the output vector with dimension $n_y = 3$, the number of state variables is $n_x = 6$ and the sample time is $T_s = 50\ \mu s$.

4.2.2.3 MPC design

Considering the identified LTI plant (4.26), the proposed MPC controller minimizes iteratively a quadratic cost function defined as

$$\min_{\Delta r} \quad \sum_{i=0}^{N-1} \|Q(y_{k+i|k} - \bar{r}_k)\|_2^2 + \|R r_{k+i|k}\|_2^2 \tag{4.27a}$$

$$\text{s.t.} \quad x_{k|k} = x(k), \tag{4.27b}$$

$$x_{k+i+1|k} = A x_{k+i|k} + B r_{k+i|k}, \tag{4.27c}$$

$$y_{k+i+1|k} = C x_{k+i+1|k}, \tag{4.27d}$$

$$i = 0, \ldots, N-1 \tag{4.27e}$$

where $Q \in \mathbb{R}^{n_y \times n_y}$ is a positive semi-definite state weighting matrix, $R \in \mathbb{R}^{n_r \times n_r}$ is a positive definite invertible control weighting matrix, N is the consider horizon, $y_{k+i|k}$ is the prediction of plant output y based on measurements available at time k, $\bar{r}_k$ is the reference signal for the controlled system at time k and $r_{k+i|k}$ is the input signal for the PI pre-compensated system described by (4.26). The minimization problem (4.27) can be cast in a parametric for-

mulation

$$\min_{z} \quad \frac{1}{2} z' H z + \rho_k' F' z \tag{4.28}$$

with $\rho_k \in \mathbb{P}$ the vector of parameters of dimension n_ρ, $\mathbb{P} \subset \mathbb{R}^{n_\rho}$ a bounded set of interest, $z \in \mathbb{R}^{n_z}$ the vector of optimization variables, $H \in \mathbb{R}^{n_z \times n_z}$ a symmetric and positive definite matrix, $F \in \mathbb{R}^{n_z \times n_\rho}$. The vector of parameters considered in the control of the closed-loop system is $\rho_k = \begin{bmatrix} r(k-1) & x(k) & \bar{r}(k) \end{bmatrix}'$ and the explicit solution z^* of the problem (4.28) is analytic and equal to $z^* = H^{-1} F \rho_k$. The proposed control scheme is presented in Fig.4.26.

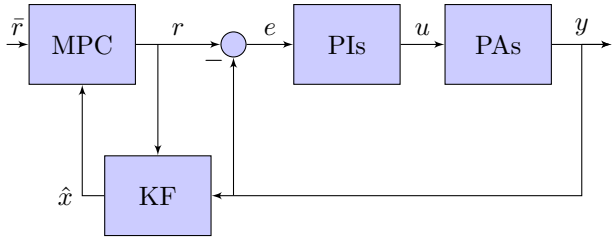

Figure 4.26: MPC+PI control scheme of the pre-compensated PAs system

4.2.2.4 Simulation and Experimental Setup

A set of simulation and experimental tests have been considered for the proposed control law. Simulations are realized in the Scilab© simulation environment in order to verify the effectiveness of the controller on the identified model. Test scenario consists into move simultaneously the system by step reference signals of amplitude 0.2 μm. The control performances are compared with respect to the results obtained by the provided analog PI controllers. The MPC controller design parameters are reported in Tab. 4.14. The parameters considered in the control performance comparison are: rise time τ_r, settling time τ_s and the Integral Square Error (ISE). Figs. 4.27 and 4.28 present simulation results comparison. In particular, Fig. 4.27 considers simulated controlled output behavior, showing the slower PI control results with respect to MPC+PI performances. Simulation results are also summarized in Tab. 4.12 showing the considered control performances indices. The MPC policy improves control results, and the faster response is related to a reduced error for each controlled axis, with an average rise time and settling improvement of 60% and 75%, respectively and an average ISE decrease of 60%. In Fig. 4.28 simulated control efforts are presented. Due to the prediction, MPC reference signals present an aggressive behavior to reduce the rise time and also to limit the output signals overshoot. Experimental results on the real PAs system confirm the performance of simulated controllers. Table 4.13 reports data of experimental

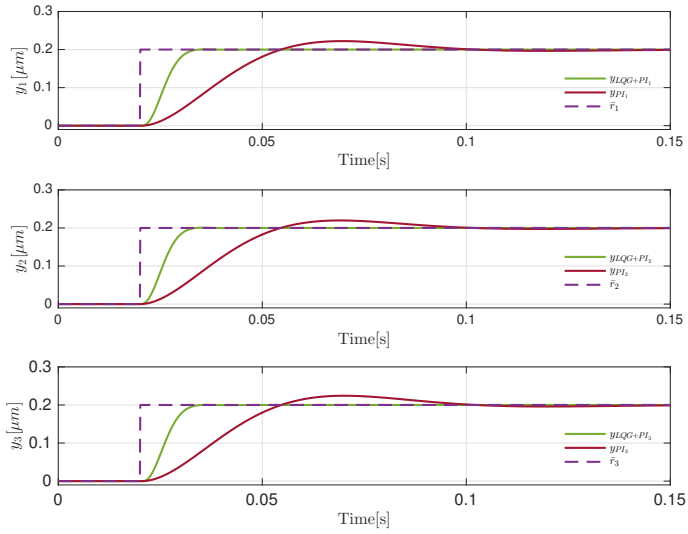

Figure 4.27: Simulation comparison between PI and MPC+PI outputs.

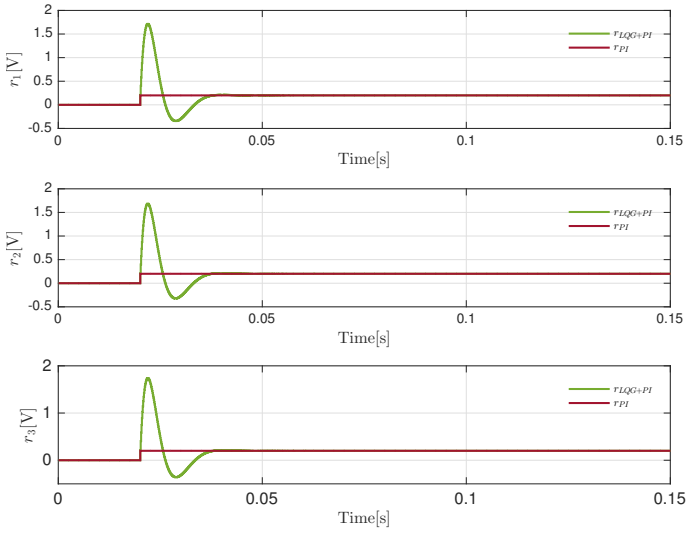

Figure 4.28: Simulation comparison between PI and MPC+PI control efforts.

step input tests. The MPC+PI controller allows to reduce the rise time of the 25% with respect to PI elapsed time. Also the settling time is improved by the MPC+PI with respect to the PI, reducing the elapsed time of the 29%. The average error computed by the ISE during the transient shows an average reduction of the 40%. Fig. 4.29 presents the controlled real plant outputs comparison. Such as in the simulation tests, the MPC+PI controlled output

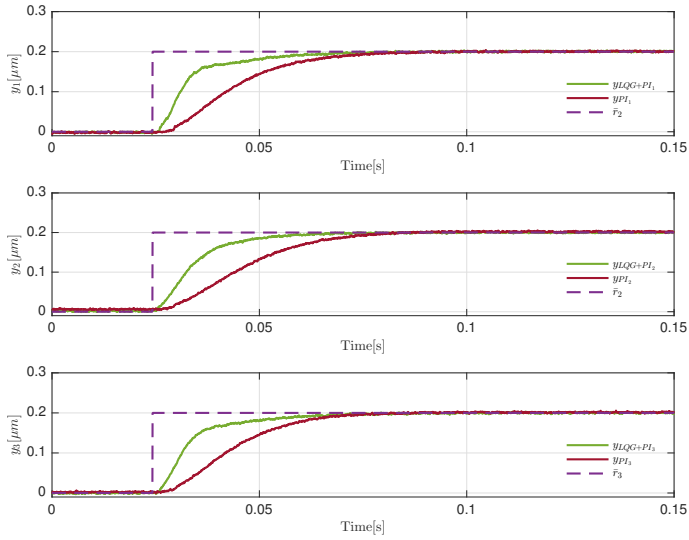

Figure 4.29: Experimental comparison between PI and MPC+PI outputs.

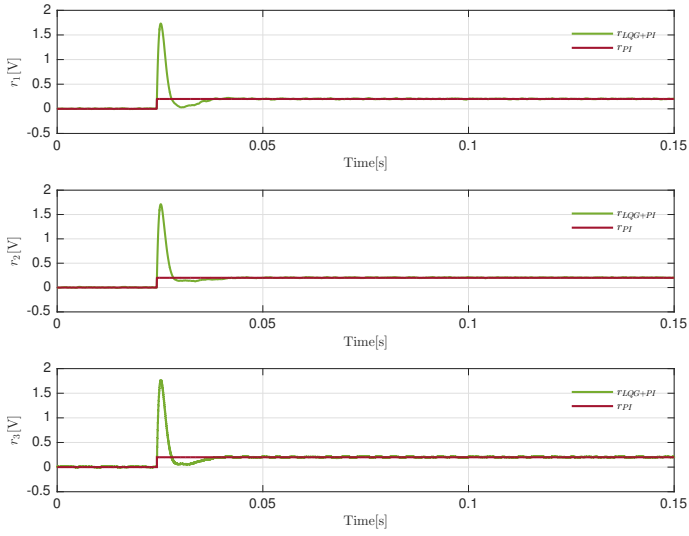

Figure 4.30: Experimental comparison between PI and MPC+PI control efforts.

is driven faster to the reference value with respect to the PIs' signal, avoiding overshoots and oscillations. In Fig. 4.30 the control efforts are reported where the MPC+PI controller presents an initial aggressive behavior as in the simulated test results.

Table 4.12: Simulation results: comparison of PI and MPC+PI under step variations

		PI	MPC+PI
X axis	τ_r[ms]	0.04	**0.02**
	τ_s[ms]	0.1	**0.025**
	ISE	0.0018	**0.00053**
Y axis	τ_r[ms]	0.04	**0.02**
	τ_s[ms]	0.1	**0.025**
	ISE	0.0017	**0.00050**
Z axis	τ_r[ms]	0.04	**0.02**
	τ_s[ms]	0.1	**0.025**
	ISE	0.0019	**0.00054**

Table 4.13: Experimental results: comparison of PI and MPC+PI under step variation

		PI	MPC+PI
X axis	τ_r[ms]	0.065	**0.048**
	τ_s[ms]	0.070	**0.050**
	ISE	0.0011	**0.00045**
Y axis	τ_r[ms]	0.065	**0.045**
	τ_s[ms]	0.070	**0.050**
	ISE	0.0012	**0.0005**
Z axis	τ_r[ms]	0.065	**0.045**
	τ_s[ms]	0.070	**0.050**
	ISE	0.0011	**0.00046**

4.3 Unmanned Vehicles

In this section different MPC control strategies have been proposed to improve the performance of Unmanned Vehicles (UVs). In Section 4.3.1 a Reference Governor control scheme has been proposed in order to improve the control performance of a pre-compensated Vertical Take-Off and Landing (VTOL) aircraft. The controlled system, modeled in the form of an LTI plant, has been provided of an external MPC control loop allowing to improve the control performance managing the inner control loop reference signals. In Section 4.3.2 a Fault Tolerant (FT) MPC policy for an over-actuated vessel has been proposed. The vessel has been modeled by a LTI system and a FT policy is introduced

Table 4.14: Controller Specifications For Simulation and Experimental Tests

Parameter	Value
MPC Parameters	
Horizon N	50
Measured variables weights Q	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Manipulated variables rate weights R	$\begin{bmatrix} 0.95 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.95 \end{bmatrix}$
Sample time T_s	50 μs

in the MPC to improve the control performances in actuators fault scenarios.

4.3.1 Reference Governor for VTOL aircrafts

In the last ten years technological progresses allowed to reduce weight, size and cost of aerial vehicles, leading to the diffusion of autonomous service aircrafts without a human pilot aboard, named Unmanned Aerial Vehicles (UAVs) or drones. Initially, commercial UAVs have been provided in rotocraft configuration: heavier-than-air flying machines that use lift generated by wings, called rotary wings or rotor blades, that revolve around a mast [91]. The success of rotocrafts was related to the simplicity of construction and maintenance, the ability to hover and the Vertical Take Off and Landing (VTOL) capability, which made them versatile systems able to operate in a larger set of scenarios with respect to classical fixed wing vehicles [92]. Recently, research community moved to study alternative drone configurations, in order to overcome rotocrafts' limitations (i.e. autonomy, range of operations and limited payload) and to extend the set of possible operating scenarios. Among different configurations, the fixed wing VTOL aircraft represents one of the more effective solutions due to the possibility to combine payload capacity, long flight range and high speed fly together with rotocrafts' features. This solution allows to employ drones for a large set of tasks in several scenarios [93–95]. On the other hands, VTOL configuration requires development of complex control system, strictly related to the considered aircraft architecture.

In this section a RG control policy for a pre-compensated VTOL aircraft is presented. The drone, presented in Section 4.3.1.1, is a fly-wing aircraft equipped with a rear propeller and two symmetric propellers on the wings. During the transition between hovering and horizontal flight, the drone is driven by a control loop composed by a Thrust Allocation (TA) algorithm and a set of primal Proportional-Integral (PI) controllers. This solution is oriented to

Table 4.15: Aircraft inertial parameters

Parameter	Symbol	Value	Unit
Mass	m	3.364	kg
Center of gravity position	x_g, y_g, z_g	$0, 0, 0$	m
Inertia moment	I_x	1.229	Kg· m^2
Inertia moment	I_y	0.1702	Kg· m^2
Inertia moment	I_z	0.8808	Kg· m^2
Inertia moment	I_{xz}	0.9343	Kg· m^2

maintain a required asset and to control directly the fly speed, in order to stabilize the aircraft. Control results are strictly related to the drone asset, due to the strong nonlinear behaviour of the system. In order to improve control performances, an external MPC loop has been introduced. The proposed MPC-based RG control has been designed with respect to an identified Linear Time Invariant (LTI) model, obtained studying the system dynamics around an appropriate operating point, as reported in Sec. 4.3.1.4. Using a linear model instead of a nonlinear representation of the system allows to reduce the computational effort required to compute the predictive control law [96] proposed in Sec. 4.3.1.5. and at the same time improves the control performance, as shown in the results given in Sec. 4.3.1.6.

4.3.1.1 VTOL aircraft dynamic model

The dynamical model of the considered drone is given in this section, describing the propellers configuration (Sec. 4.3.1.2) and the primal controller (Sec. 4.3.1.3). The LTI model considered in the RG control is reported in Sec. 4.3.1.4. The comparison between primal controllers and RG-based solution has been reported in Sec. 4.3.1.6.

4.3.1.2 Mathematical model

The dynamical model of the drone has been developed as proposed in [97, 98] and simulated by the Matlab® Aerosim blockset provided by u-dynamics [99].

The vehicle is provided by a rear propeller and two symmetric wing propeller to generate the vertical take off thrust. The propulsion system has been developed studying the propellers thrust scheme reported in Fig. 4.31. Each propeller is composed by a counter-rotating double helical, allowing to improve the propeller thrust and avoiding to generate a moment of force with respect to the body frame z axis. Rear propeller, posed at $(-0.339, 0, 0)$[m, m, m] with respect to the origin of the body frame O_B, provides a thrust F_{MP} parallel to the vehicle x_B axis. Wings propellers are posed symmetrically with respect to O_B

Table 4.16: Aircraft aerodynamic parameters

Parameter	Symbol	Value	Unit
Wingspan	s_p	2.1	m
Wing surface	s_u	0.75	m^2
Wing chord	$\bar{c}$	0.3571	m
Oswald coefficient	e	0.9935	
Transition Rate	M	50	
Stall angle	α_0	0.2670	rad
Lift coeff.	C_{L_α}	4.0191	
	C_{L_0}	0.0254	
	C_{L_q}	3.8954	
	$C_{L_{\delta_e}}$	0.5872	
Drag coeff.	$C_{D_{\delta_e}}$	0.8461	
	C_{D_p}	0.0102	
	$C_{D_{\beta_2}}$	0.0671	
	$C_{D_{\beta_1}}$	$-2.0864e^{-7}$	
	$C_{D_{\beta_0}}$	$7.7235e^{-5}$	
	C_{D_q}	0	
Pitch coeff.	C_{m_α}	-0.2524	
	C_{m_0}	0.0180	
	$C_{m_{\delta_e}}$	-0.4857	
	$C_{m_{fp}}$	-0.2168	
	C_{m_q}	-1.3047	
Side coeff.	C_{Y_0}	$3.2049e^{-18}$	
	C_{Y_p}	-0.1172	
	C_{Y_r}	0.0959	
	$C_{Y_{\delta_a}}$	-0.0696	
	$C_{Y_{\delta_r}}$	0	
	C_{Y_β}	-0.1949	
Roll coeff.	C_{l_0}	$1.1518e^{-18}$	
	C_{l_p}	-0.4018	
	C_{l_r}	0.0250	
	$C_{l_{\delta_a}}$	0.2987	
	$C_{l_{\delta_r}}$	0	
	C_{l_β}	-0.0765	
Yaw coeff.	C_{n_0}	$-2.2667e^{-7}$	
	C_{n_p}	-0.0247	
	C_{n_r}	-0.1252	
	$C_{n_{\delta_a}}$	0.0076	
	$C_{n_{\delta_r}}$	0	
	C_{n_β}	0.0403	

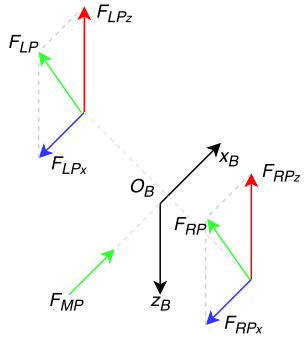

Figure 4.31: Propellers' force diagram

at $(-0.11449, \pm 0.525, 0)$[m, m, m], with an inclination of $\gamma_{RP} = \gamma_{LP} = 0.0174$ rad with respect to the normal to the plan described by the x_B and z_B axis of the aircraft body frame. This angle allows to decompose wings' propellers forces F_{LP}, F_{RP} in two components with respect to the x_B axis (F_{LP_x}, F_{RP_x}) and the z_B axis (F_{LP_z}, F_{RP_z}). Then the vertical components F_{LP_z}, F_{RP_z} provide the required thrust to take-off the drone and the rear propellers force F_{MP} compensates the contribution of the horizontal components F_{LP_x}, F_{RP_x}, allowing to have a vertical take-off without horizontal drifts. Propellers' configuration allows the drone to vertically take-off and landing as a rotocraft system. Furthermore, the drone has been designed also to fly horizontally as a fixed-wing aircraft, and an ad-hoc controller managing the system during the transition between the flight configurations has been considered. The target of this controller is to stabilize the drone asset during the transition, guaranteeing to effectively control the drone speed, in order to reach and maintain critical flight speed with a prescribed asset. Due to the drone architecture, the full asset can be controlled driving the pitch and roll of the system. The roll and yaw angles dynamics are coupled, and have to be stabilized together. Forces $f_P = \begin{bmatrix} F_{MP} & F_{LP} & F_{RP} \end{bmatrix}'$ and moments $M_P = \begin{bmatrix} M_{MP} & M_{LP} & M_{RP} \end{bmatrix}'$ provided by rear, left and right propellers respectively are

$$f_P = \frac{4}{\pi^2}\rho \begin{bmatrix} R_{MP}^4 \Omega_{MP}^2 C_{T_{MP}} \\ R_{LP}^4 \Omega_{LP}^2 C_{T_{LP}} \\ R_{RP}^4 \Omega_{RP}^2 C_{T_{RP}} \end{bmatrix}, \; M_P = \begin{bmatrix} r_M \times F_{MP} \\ r_L \times F_{LP} \\ r_R \times F_{RP} \end{bmatrix} \tag{4.29}$$

where $r_M = 0.18$ [m] and $r_L = r_R = 0.15$ [m] are the radius of main, left and right propeller, respectively. From Eq. (4.29), the resulting propellers forces $F_P = \begin{bmatrix} F_{P_x} & F_{P_y} & F_{P_z} \end{bmatrix}'$ and aerodynamic forces $F_A = \begin{bmatrix} F_{A_x} & F_{A_y} & F_{A_z} \end{bmatrix}'$,

with respect to the axis of the body reference frame are

$$F_P = \begin{bmatrix} F_{MP} + F_{LP}\cos\gamma_{LP} + F_{RP}\cos\gamma_{RP} \\ 0 \\ -F_{LP}\sin\gamma_{LP} - F_{RP}\sin\gamma_{RP} \end{bmatrix}$$
$$F_A = qS \begin{bmatrix} -C_D\cos\alpha + C_L\sin\alpha \\ C_Y \\ -C_D\sin\alpha - C_L\cos\alpha \end{bmatrix} \tag{4.30}$$

and aerodynamic moments are

$$M_A = \begin{bmatrix} M_{A_x} \\ M_{A_y} \\ M_{A_z} \end{bmatrix} = qs_u \begin{bmatrix} s_p C_L \\ \bar{c} C_m \\ s_p C_E \end{bmatrix} \tag{4.31}$$

where q is the dynamic pressure and the terms C_X, C_Y, C_Z, C_l, C_m, C_E are computed from the aerodynamics coefficient of Tab. 4.16 according with [98, 100]. Considering (V, U, W) the translational speeds, (P, Q, R) the angular velocities, (X, Y, Z) the forces and (L, M, N) the moments with respect to the x, y and z axis respectively, the dynamical model of the proposed aircraft is

$$\begin{aligned} X &= m\left(\dot{V} + QW - RU + g\sin\theta\right) + F_{P_x} + F_{A_x} \\ Y &= m\left(\dot{U} + VR - WP - g\cos\theta\sin\psi\right) + F_{P_y} + F_{A_y} \\ Z &= m\left(\dot{W} + UP - QV - g\cos\theta\cos\psi\right) + F_{P_z} + F_{A_z} \\ L &= I_x\dot{P} - I_{xz}\left(\dot{R} + PQ\right) + \left(I_z - I_y\right)QR + M_{P_x} + M_{A_x} \\ M &= I_y\dot{Q} + I_{xz}\left(P^2 - R^2\right) + \left(I_x - I_z\right)PR + M_{P_y} + M_{A_y} \\ N &= I_z\dot{R} - I_{xz}\dot{P} + \left(I_y - I_x\right)PQ + I_{xz}QR + M_{P_z} + M_{A_z}. \end{aligned} \tag{4.32}$$

The inertial parameters of the drone are reported in Tab. 4.15 and the aerodynamic parameters are grouped in Tab. 4.16.

4.3.1.3 Primal controller

The primal control loop scheme given in Fig. 4.32 has been designed considering a sample time T_p. It is composed by a thrust allocation (TA) policy that coordinates the actuators, and a set of proportional-integral (PI) regulators to stabilize the system asset and driving the vehicle speed. This controller has been provided in order to guarantee that in the considered flight scenario *i*) propellers' force compensate vehicle weight; *ii*) drone speed is controlled; *iii*) drone asset is stabilized. In the following the TA algorithm is reported, introducing the formulation with respect to the control target, and the asset and speed regulators are given, considering the control scenario.

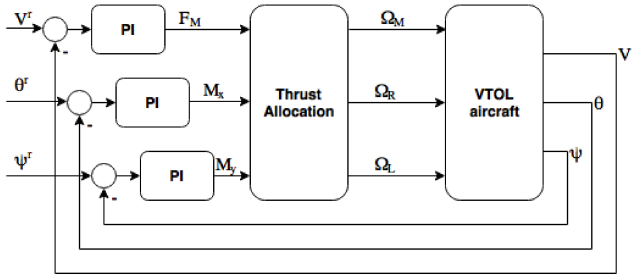

Figure 4.32: Block scheme of the primal controller.

4.3.1.3.1 Thrust Allocation policy The thrust allocation algorithm allows to optimally map the PI control effort $\tau = \begin{bmatrix} F_M & M_x & M_y \end{bmatrix}'$ where F_M is the speed control force and M_x and M_y are the x and y body frame axis moments of force, respectively. The available actuators' effort is $f = \begin{bmatrix} \Omega_M & \Omega_R & \Omega_L \end{bmatrix}'$ where Ω_M, Ω_R and Ω_L are the rotational speed of main, right and left propeller, respectively. The TA algorithm has been formulated as a quadratic programming (QP) problem [101], with constraints on the maximum and minimum rotation speed of the propellers, i.e. $f_{max} = \begin{bmatrix} \Omega_M^{max} & \Omega_R^{max} & \Omega_L^{max} \end{bmatrix}'$ and $f_{min} = \begin{bmatrix} \Omega_M^{min} & \Omega_R^{min} & \Omega_L^{min} \end{bmatrix}'$. In particular the TA algorithm is given by

$$\begin{aligned} &\min_{f,s,\bar{f}} \left\{ f'Wf + s'Qs + \beta\bar{f} \right\} \\ &\text{s.t.} \quad Tf = \tau + s \\ &\qquad f_{min} \leq f \leq f_{max} \\ &\qquad -\bar{f} \leq f_1, f_2, f_3 \leq \bar{f} \end{aligned} \tag{4.33}$$

where W is a positive definite weight matrix, $s'Qs$ is the penalty term for the slack variables vector s, the last term βf, with $\beta \leq 0$, has been introduced in order to minimize the bigger force $\bar{f} = \max_i f_i$ with $i = 1, 2, 3$. T is the configuration matrix defined from the aircraft layout

$$T = \begin{bmatrix} K_M & 0 & 0 \\ 0 & -l_{L_Y} K_L \sin\theta_L & -l_{R_Y} K_R \sin\theta_R \\ 0 & -l_{R_X} K_L \cos\theta_L & -l_{R_X} K_R \cos\theta_R \end{bmatrix} \tag{4.34}$$

with

$$\begin{aligned} K_M &= \frac{4}{\pi^2}\rho r_M^4 C_{T_M} \\ K_L &= \frac{4}{\pi^2}\rho r_L^4 C_{T_L} \\ K_R &= \frac{4}{\pi^2}\rho r_R^4 C_{T_R} \end{aligned} \tag{4.35}$$

and where l_{i_j} (with $i = M, R, L$ for main, right and left propeller, respectively and $j = x, y, z$ for the related axis) is the arm of actuator force i with respect to the axis j, r_i is the radius of the propellers i and C_{T_i} is the moment coefficient of the propeller i.

4.3.1.3.2 PI Regulators The considered low level controllers are PI regulators to control the horizontal speed and stabilize the asset of the drone. The control performance are strictly related to the drone asset and they can undergo severe degradation in presence of unconsidered coupling effects between different outputs. PI controllers have been tuned consider the drone dynamics decouple, and stabilizing the vehicle to the flight configuration given by $\theta = \psi = 0$ [rad], $V = 0.1$ [m/s].

4.3.1.4 Identification of pre-compensated drone models

The nonlinear aircraft system driven by the primal controller has been modelled with a LTI system obtained by an identification task. Considering the operating point $V_0 = 0.1$ m/s, $\psi_0 = 0.08$ rad, $\theta_0 = 0.4363$ rad, input-output data of the closed loop plant subjected to reference signals variations ($|\Delta V^r| \leq 0.05$ m/s, $|\Delta\psi^r| \leq 0.04$ rad, $|\Delta\theta^r| \leq 0.22$ rad) have been collected with a sample time $T_r = 0.2$ s. The collected data have been used with a subspace identification technique [102], in order to obtain a discrete time LTI model in space-state representation

$$\begin{aligned} x_{k+1} &= Ax_k + Br_k \\ y_k &= Cx_k \end{aligned} \tag{4.36}$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ and $C \in \mathbb{R}^{n_y \times n_x}$ are the space-state matrices, $x_k \in \mathbb{R}^{n_x}$ is the state array with $n_x = 3$, $r_k = \begin{bmatrix} V_k^r & \psi_k^r & \theta_k^r \end{bmatrix}'$ is the input array of dimension $n_u = 3$ and $y_k = \begin{bmatrix} V_k & \psi_k & \theta_k \end{bmatrix}'$ is the output array of dimension $n_y = 3$. The model of Eq. (4.36) has been validated comparing the data collected from the nonlinear plant with respect to the signals generated by the identified model. The resulting average fit (drone speed 77.34%, pitch 85, 98% and roll 70.69%) validates the model fidelity, showing the goodness of

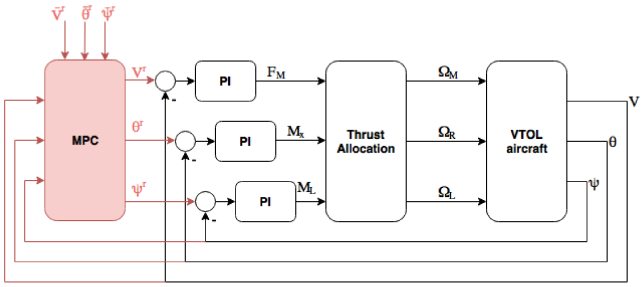

Figure 4.33: Block scheme of the Reference Governor.

the identification procedure. The identified LTI model has been used to develop the predictive controller considered in the proposed RG control policy.

4.3.1.5 Reference Governor

The proposed RG control policy considers the identified LTI model of Eq. (4.36) to develop a MPC to manage the primal controller reference signals. The reference managing control loop is reported in the scheme of Fig. 4.33. When the cost function is quadratic and the constraints are affine the predictive control optimization problem to be solved at each time step can be formulated as

$$\min_{\Delta u} \quad \sum_{i=1}^{N_p} \left(\|W_y(y_{k+i|k} - \bar{r}_k)\|_2^2 + \|W_u r_{k+j|k}\|_2^2 \right) \sum_{j=0}^{N_u-1} \|W_{\Delta u} \Delta r_{k+j|k}\|_2^2 \tag{4.37a}$$

$$\text{s.t.} \quad x_{k+i+1|k} = Ax_{k+i|k} + Br_{k+i|k} \tag{4.37b}$$

$$y_{k+i|k} = C_k x_{k+i|k} \tag{4.37c}$$

$$x_{k|k} = x_k \tag{4.37d}$$

$$\Delta r_{k+i|k} = r_{k+i|k} - r_{k+i-1|k} \tag{4.37e}$$

$$\Delta r_{k+N_u+h|k} = 0 \tag{4.37f}$$

$$r_{k+j|k} \in \mathbb{U} \tag{4.37g}$$

$$y_{k+i|k} \in \mathbb{Y} \tag{4.37h}$$

$$i \in \{1, \dots, N_p\} \tag{4.37i}$$

$$j \in \{0, \dots, N_u - 1\} \tag{4.37j}$$

$$h \in \{0, \dots, N_p - N_u - 1\} \tag{4.37k}$$

where N_p is the prediction horizon, N_u is the control horizon, W_y, W_u, $W_{\Delta u}$ are square weight matrices of appropriate dimension, with $W_{\Delta u}$ invertible, $x_{k+i|k}$ denotes the prediction of the state vector x at time $k+i$ based on the information available at time k, $\Delta u_{k+i|k}$ is the vector of the input increments, $\bar{r}_k$ is the vector of output references and $\mathbb{U}$, $\mathbb{Y}$ are polyhedral sets of constraints inputs and outputs respectively. The problem (4.37) can be cast into the parametric Quadratic Programming (QP) problem

$$\begin{aligned} \min_{z} \quad & \frac{1}{2} z' H z + \rho_k' F' z \\ \text{s.t.} \quad & Gz - w - W\rho_k \leq 0 \end{aligned} \tag{4.38}$$

with $\rho \in \mathbb{P}$ the parameters vector, $\mathbb{P} \in \mathbb{R}^{n_\rho}$ a bounded set of interest, $z \in \mathbb{R}^n$ the vector of optimization variables, $H \in \mathbb{R}^{n \times n}$ a symmetric and positive definite matrix, $F \in \mathbb{R}^{n \times n_\rho}$, $G \in \mathbb{R}^{m \times n}$, $w \in \mathbb{R}^m$ and $W \in \mathbb{R}^{m \times n_\rho}$. Furthermore, the proposed MPC has been developed with respect to the LTI model of Eq. (4.36) that does not consider the non-linearity of the plant. In order to compensate errors related to model uncertainties an integral control action is introduced in the control law, modifying the reference $\bar{r}_k$ considered by the MPC with

$$\tilde{r}_k = \tilde{r}_{k-1} + k_r(\bar{r}_k - y_k) \tag{4.39}$$

where $\tilde{r}_k = \begin{bmatrix} \tilde{V}_k^r & \tilde{\psi}_k^r & \tilde{\theta}_k^r \end{bmatrix}'$, $\bar{r}_k = \begin{bmatrix} \bar{V}_k^r & \bar{\psi}_k^r & \bar{\theta}_k^r \end{bmatrix}'$ and $k_r \in \mathbb{R}^{n_y \times n_y}$ is a diagonal matrix of integral gains k_V, k_ψ and k_θ tuned in calibration, such that

$$k_r = \begin{bmatrix} k_V & 0 & 0 \\ 0 & k_\psi & 0 \\ 0 & 0 & k_\theta \end{bmatrix}. \tag{4.40}$$

A Kalman filter [103] has been introduced in order to obtain an estimation of the plant state vector. The proposed predictive controller structure is reported in Fig. 4.34.

4.3.1.6 Simulation Results

In this section simulation results have been reported in order to compare the primal controller with the RG control performance. The algorithm considered to solve the TA and MPC constrained optimization problems is the MATLAB™active-set based QP solver quadprog. Tab. 4.17 collects the primal controller and MPC tuning parameters. Fig. 4.35 reports the control performance provided by the primal controller and the proposed RG control policy. Red dashed line represents the original reference $\bar{r}_k$ for the external RG control loop, the black dashed line is the managed reference r_k computed by the MPC,

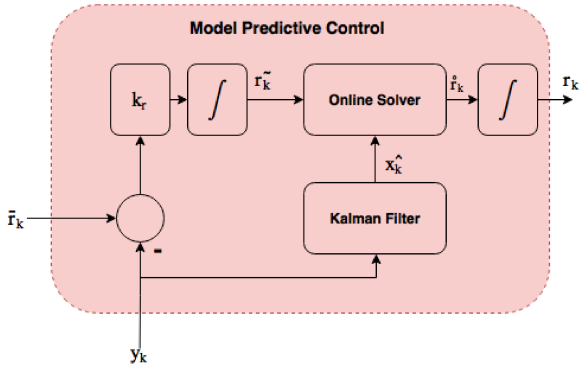

Figure 4.34: Block scheme of the MPC.

the green line is the output controlled by the primal controller y_{PI} and the blue line is the output driven by the introduced RG control policy y_{RG}. Fig. 4.36 shows the PI control effort $\tau^{PI} = \begin{bmatrix} F_M^{PI} & M_x^{PI} & M_y^{PI} \end{bmatrix}'$ (red line) when only the primal controller is considered with respect to the control effort obtained introducing the RG control, i.e. $\tau^{RG} = \begin{bmatrix} F_M^{RG} & M_x^{RG} & M_y^{RG} \end{bmatrix}'$ (black dashed line).

The introduction of the slower MPC external control loop allows to improve the control performance, reducing the transient overshot and the steady state offset errors. These issues are related to the nonlinearities and to the coupling effects between the controlled outputs, that reduce primal control loop performance. The external MPC allows to address these problems and to improve control performance managing the reference for the primal controllers. In particular the asset control performance shows a strong improvement due to the primal controller signals managing. Overshoot, oscillations and steady-state errors of pitch and roll angles have been compensated and simultaneously the RG control allows to obtain faster and smoother transient responses.

Furthermore, also the primal control effort provided by the PI controllers is improved. Due to the introduction of the RG control, the PI control efforts, i.e. the reference for the TA algorithm, present a smooth behavior with reduced oscillations and peaks, which is equivalent to require less effort from system actuators and therefore less power consumption. On the contrary, the speed dynamic is featured by a faster response with respect to the other controlled variables, and the slower MPC external loop can not introduce a substantial performance improvement.

In order to verify the goodness of the proposed controller, also a set of random variable reference signals have been considered. In fact control performance are strictly related to the drone asset, that introduced strong variations in the sys-

Table 4.17: TA, PI and MPC Tuning Parameters

Parameter	Symbol	Value	Units
Primal sampling time	T_p	0.02	s
Max. propellers' speed	f_{max}	$\begin{bmatrix} 2.72e03 \\ 2.72e03 \\ 2.72e03 \end{bmatrix}$	rad/m
Min. propellers' speed	f_{min}	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	rad/m
Propellers' weights	W	$\begin{bmatrix} 0.002 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$	
Slack variable weights	Q	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot 1e16$	
Speed proportional gain	k_p^V	280	
Speed integral gain	k_i^V	150	
Pitch proportional gain	k_p^θ	50	
Pitch integral gain	k_i^θ	10	
Roll proportional gain	k_p^ψ	8	
Roll integral gain	k_i^ψ	10	
RG sampling time	T_r	0.2	s
Prediction horizon	N_p	10	
Control horizon	N_u	3	
MO weights	Q_y	$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$	
MV weights	Q_u	$\begin{bmatrix} 6 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$	
MV rate weights	$Q_{\Delta u}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$	
Speed constraints	V^{max}	10	m/s
	V^{min}	-1	m/s
Pitch constraints	θ^{max}	0.78	rad
	θ^{min}	0.78	rad
Roll constraints	ψ^{max}	0.78	rad
	ψ^{min}	0.78	rad
Reference integral gains	k_V	0.05	
Reference integral gains	k_ψ	0.7	
Reference integral gains	k_θ	0.03	

Table 4.18: Reference governor average percentage performance improvement

	$\Delta\text{ISE}_{\%}$	$\Delta\text{IAE}_{\%}$	$\Delta\text{ITAE}_{\%}$	$\Delta\tau_{r\%}$	$\Delta\tau_{s\%}$
V	10.14	13.98	11.63	5.18	7.23
ψ	58.35	65.59	63.13	16.25	23.13
θ	49.19	49.28	50.13	23.52	35.82

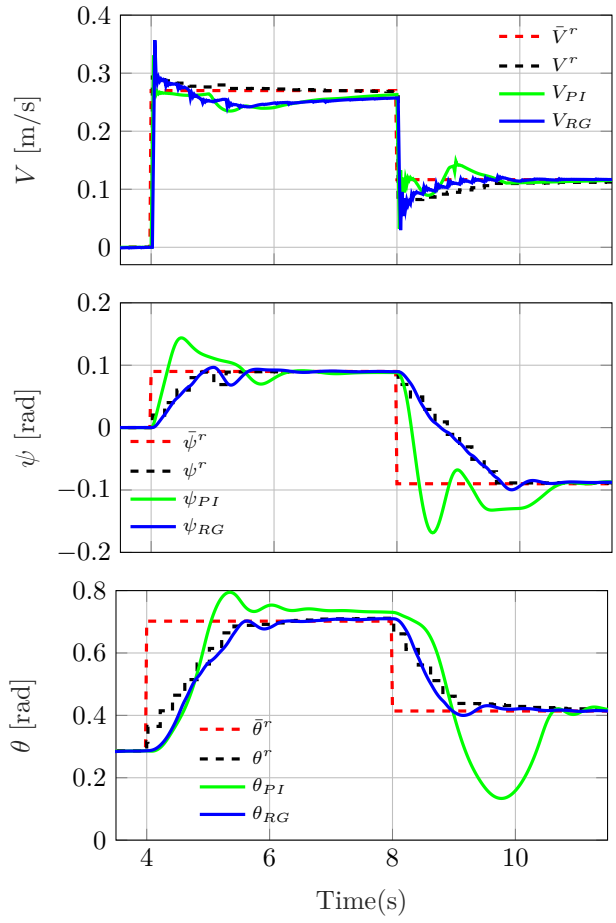

Figure 4.35: Scenario 1: comparison of the primal controller and the RG algorithms when controlling the VTOL aircraft.

tem dynamics. Considering a large set of sequences of variable reference signals (inside the space of values given by the MPC constrains), the control results have been evaluated in terms of the average percentage improvement of the Integral Square Error $\Delta\text{ISE}_{\%}$, the Integral Absolute Error $\Delta\text{IAE}_{\%}$, the Integral

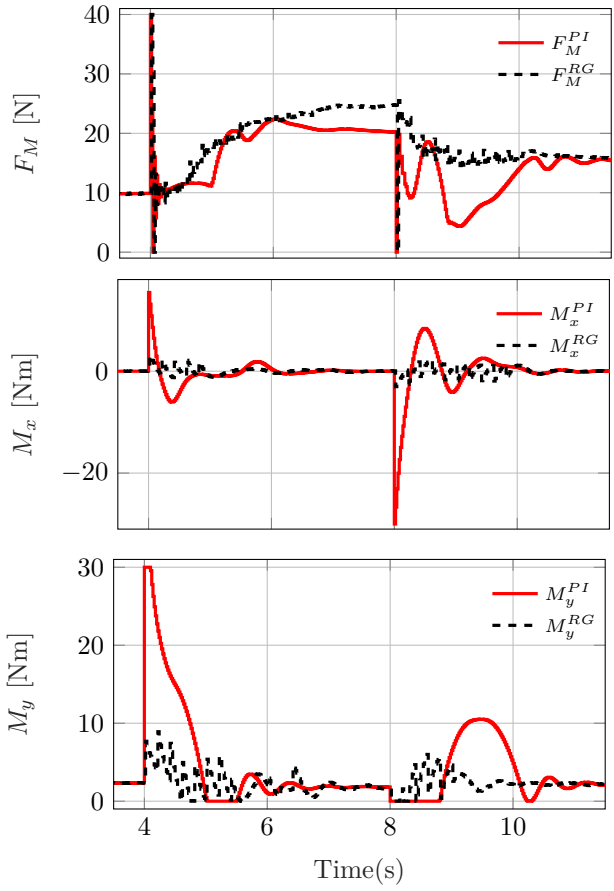

Figure 4.36: Comparison of the primal controller and the RG algorithms PI control efforts when controlling the VTOL aircraft.

Time-weighted Absolute Error $\Delta\text{ITAE}_{\%}$, the rise time $\Delta\tau_{r\%}$ and the settling time $\Delta\tau_{s\%}$. These results are summarized in Tab. 4.18 and shown the improvement introduced from the RG control policy, that allows to reduce both the average settling and rise times, and to decrease the average percentage of the ISE, IAE and ITAE values. These results confirm the goodness of the introduced controller, able to guarantee a considerable performance improvement considering drone assets configurations allowed by the MPC constraints.

4.3.2 Fault Tolerant MPC for Over-actuated vessels

A Fault Tolerant Control System (FTCS) is a closed loop controller able to automatically accommodate component failures, maintaining system stability and acceptable control performance in the event of fault [104]. Research and

development of FTCS and Fault Detection and Diagnosis (FDD) systems involved large part of the international research community, due to a growing demand for safety, reliability, maintainability, and survivability in dynamical systems [105]. FDD systems detect if there is a fault in the system and provide some specific information about it [106]. On the other hand, the FTCS is designed to use information provided by the FDD to re-modulate the controller depending on the availability of redundancies in the control system as well as design approaches used in the synthesis of fault-tolerant controllers [107]. During the past few years Model Predictive Control (MPC) has been proposed in conjunction with FDD techniques in order to develop FT controllers [108,109]. In this thesis a Fault Tolerant (FT) MPC policy has been proposed to control an over-actuated vessel model. The considered plant is the Cybership II [110], a scale model of a vessel provided with a redundant set of actuators. For this reason it requires a low level Thrust Allocation (TA) algorithm in order to obtain prescribed closed loop control performance by an optimal control force allocation among available actuators. The introduction of TA policies is a standard solution to simplify the design of control system for over-actuated system and it is usually considered to develop FTCS [111,112].

The main contributions of the proposed solution are *i*) the development of a Linear Time-Invariant (LTI) model of the Cybership II used to design the considered MPC; *ii*) the introduction of a feedback loop between the TA algorithm and the MPC, in order to model possible actuators' fault or failure and to automatically re-modulate the fault-free control law by the introduction of the fault effect into the MPC reference model. Due to the considered feedback, the FT-MPC allows to improve control performances in actuators' fault scenarios and guarantees system safety in case of critical faults combinations. The proposed solution has been tested in different actuators' failure scenarios, comparing control results with standard MPC performances by the analysis of a standard control index and showing the transient response improvement in the reported simulated results. Due to the presence of wave disturbances, in real case the control performance can be effected by the presence of an output additive noise [113,114]. In order to verify control performance also in case of strong additive noise, comparison results varying the intensity of the additive output wave disturbances have been reported.

This part of the thesis is organized as follows. In Section 4.3.2.1 the vessel dynamic model has been reported and in Sec. 4.3.2.2 the thrust allocation policy is given. In Sec. 4.3.2.3 the LTI model of the vessel is reported. The MPC policy and the FT-MPC scheme have been designed in Sec. 4.3.2.3.2 and Sec. 4.3.2.5 shows the control result comparison.

4.3.2.1 Cybership II dynamic model

The Cybership II [110] is a scale (1:70) model of a over-actuated supply vessel having a mass of 23.8 kg, a length of 1.255 m, a breadth of 0.29 m. The ship has two main propellers, two rudders aft and one bow thruster. The dynamic model is fully described by six degree-of-freedom (DOF) related to the ship position in the three-dimensional space (X, Y and Z positions called surge, sway, and heave, respectively) and orientation (ϕ, θ and ψ angles called roll, pitch and yaw, respectively). In the following the ship has been considered longitudinally and laterally metacentrically stable due to the assumption of small amplitude of roll and pitch angles and angle rate, i.e. $\phi = \theta = \Delta\phi = \Delta\theta \approx 0$. In the same way it is possible to discard the heavy dynamic considering the ship floating to $z \approx 0$. The result of the previous assumptions is that the Cybership II model is described by a 3 DOF equation system related to the state array $\eta = [x, y, \psi]'$ referred to the North-East-Down (NED) earth-fixed reference frame. Considering $\upsilon = [u, v, r]'$, with u and v the surge and sway velocities and r the yaw rate, the ship rigid body dynamic is described by the kinematic relationship:

$$\dot{\eta} = R(\psi)\upsilon \tag{4.41}$$

where $R(\psi)$ is the rotation matrix defined as:

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.42}$$

As proposed in [115] the ship can be considered a low-speed surface vessel, described by:

$$\begin{aligned} \dot{\eta} &= R(\psi)\upsilon \\ M\dot{\upsilon} &= -D\upsilon + \bar{u} \end{aligned} \tag{4.43}$$

with $\bar{u} = [\tau_X, \tau_Y, \tau_N]'$ the input vector, where τ_X and τ_Y are the surge and sway controls force, τ_N is the yaw control moment and M, D are the mass matrices and damping, respectively:

$$\begin{aligned} M &= \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & x_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \\ D &= \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -Y_r & -N_r \end{bmatrix}. \end{aligned} \tag{4.44}$$

Table 4.19: Cybership II dynamical model parameters

Parameter	Symbol	Value	Unit
Mass	m	23.8	kg
CG position	x_g	0.0425	m
Inertia moment	I_z	1.76	Nm
Added mass derivate	$X_{\dot{u}}$	-2.0	kg
Added mass derivate	$Y_{\dot{v}}$	-100	kg
Added mass derivate	$Y_{\dot{r}}$	0.0	kgm
Added mass derivate	$N_{\dot{r}}$	-1.0	kgm^2
Damping force coef.	X_u	-2.0	kg/s
Damping force coef.	Y_v	-7.0	kg/s
Damping moment coef.	N_r	-0.5	kgm^2/s

The Cybership II dynamical model parameters have been reported in Tab.4.19.

4.3.2.2 Identification of the Cybership II model

A Linear Time-Invariant (LTI) model of the nonlinear vessel dynamics has been considered. The nonlinear plant (4.43) of the Cybership II has been identified through the analysis of the input/output data obtained by several simulation tests. Data have been acquired with a sample rate of 4 Hz and a set of bounds is considered for the generated input data:

$$|\tau_X| \leq 2N \ , \ |\tau_Y| \leq 1.5N \ , \ |\tau_N| \leq 1.5kg\ m^2/s. \tag{4.45}$$

Control inputs have been applied to the vessel by the unconstrained TA algorithm discussed in Sec. 4.3.2.4. The considered plant output vector $\bar{z}$ is composed by the ship positions ($x(k)$ [m], $y(k)$ [m]) and orientation ($\psi(k)$ [rad]), referred to the NED external fixed reference system. The Wave Noise Filter (WF) proposed in [116] has been considered to estimate the true ship asset filtering the disturbances $d(k)$, due to the sea wave. Both identification and validation dataset have been acquired introducing an output additive wave noise described by

$$h_w^i(s) = \frac{\sigma^i s}{s^2 + \zeta_i \omega_{0_i} s + \omega_{0_i}^2} \tag{4.46}$$

where $i = 1, 2, 3$ is the index relative to the considered output (surge, sway and yaw, respectively), $\sigma^i = 0.5$ is the parameter related to the wave intensity, $\zeta_i = 0.1$ is the relative damping ratio and $\omega_{0_i} = 0.8976$ rad/s is the dominating wave frequency in order to generate a wave disturbance featured by a frequency of 0.1429 Hz. In the identification procedure the open loop system

Table 4.20: Identified model NRMSE validation results

N_p	x[%]	y[%]	ψ[%]
1	100	100	100
10	99.95	99.93	99.93
20	99.78	99.67	99.74
50	98.77	98.02	98.81
100	96.32	94.13	97.64

has been forced with a set of random steps with a fixed period of 400 s in order to record the nonlinear plant steady state response. During the data acquisition the plant inputs have been forced together in order to model any possible coupling effects between inputs and outputs. The collected input and output data have been used to identify a LTI model by a least square identification algorithm. A sample of the result of the LTI model validation is reported in Fig. 4.37 where the nonlinear plant outputs behavior has been compared with the outputs computed by the identified model, showing the effectiveness of the linear plant to reproduce the Cybership II dynamic. The identified model has been tested comparing the computed outputs with respect to several validation data collected from the nonlinear model. Validation tests results have been evaluated in terms of the Normalized Root Mean Square Error (NRMSE) varying the considered prediction horizon N_p as reported in Tab. 4.20.

The result of the identification is a discrete time LTI model $\mathcal{S}$

$$\begin{aligned} \bar{x}(k+1) &= A\bar{x}(k) + B\bar{u}(k) \\ \bar{z}(k) &= C\bar{x}(k) \end{aligned} \tag{4.47}$$

where $\bar{u}(k) = [\tau_X(k), \tau_Y(k), \tau_N(k)]'$ is the input vector of dimension $n_{\bar{u}} = 3$, $\bar{z}(k) = [x(k), y(k), \psi(k)]'$ is the output vector of dimension $n_{\bar{z}} = 3$. The state vector $\bar{x}(k)$ has dimension $n_{\bar{x}} = 10$ and the sample time is $T_s = 0.25$ s. The identified system of Eq. (4.47) is featured by a full rank diagonal dominant state transition matrix A, that allow to represent the non linear plant state evolution of Eq. (4.43) by a constant matrix instead of the time-variant matrix $R(\psi)$ that requires to be recomputed iteratively with respect to the instantaneous value of the yaw angle ψ_k.

4.3.2.3 Controller structure

The architecture of the controller is shown in Fig. 4.38. It is composed of a control allocation (TA) algorithm, used to assign the nominal control effort $\bar{u}$ among the available actuators and providing the real thrust effort $\bar{u}_r$; a wave

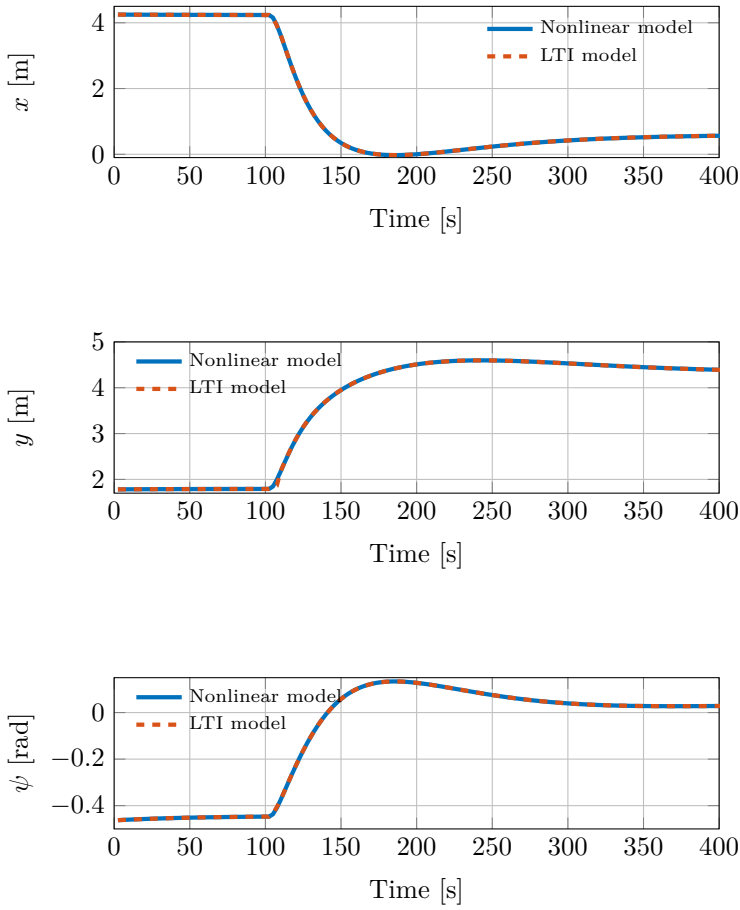

Figure 4.37: Validation results of the identified Cybership II LTI model. From top to bottom: the nonlinear model x, y axis position and the ψ angle (blue) and the LTI model model output (red), respectively.

filter (WF) that allows to estimate the ship position $\bar{y}_r$ in presence of external environment noise d affecting the measured output $\bar{y}_m$; the controller (MPC) tuned with respect to the identified model (4.47), with the feedback signals f_r and f_* from the TA in order to compute the considered fault tolerant policy.

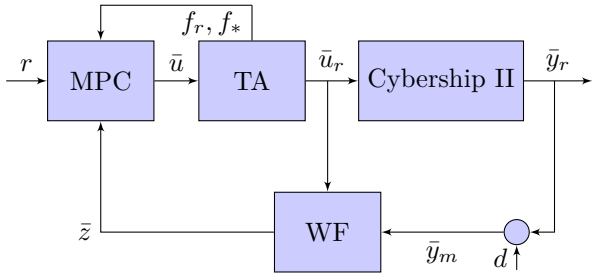

Figure 4.38: Architecture of the fault tolerant model predictive controller

4.3.2.3.1 Thrust Allocation policy The control allocation policy coordinates the different thrusts of the over-actuated system in order to produce the desired control efforts. The TA module architecture is reported in Fig.4.39. The TA algorithm allocates the required control effort $\bar{u}$ among the provided actuators set with respect to actuators position and orientation. The TA solution is the allocated control force f_* that represents the reference signal for the real actuators. The actuators set driven by f_* provides the real force signals f_r used to compute the real control effort $\bar{u}_r$ by the thrust inverse mapping related to the geometrical configuration of the vessel actuators. Several TA algorithms have been proposed in last years and for the state of the art of control allocation policies it is possible to refer to [101].

A Quadratic Programming (QP) formulation of the unconstrained TA solution for the fixed actuators allocation problem has been introduced in the control loop of the Cybership II. Given the actuators vector $p \in \mathbb{R}^{n_p}$ such that

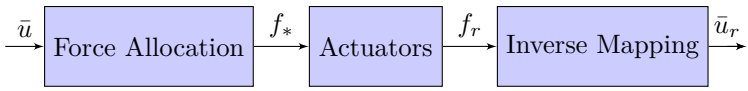

Figure 4.39: Thrust allocation algorithm block scheme

$n_p = n_{p_f} + n_{p_r}$, with n_{p_f} fixed and n_{p_r} rotable actuator, respectively, each fixed actuator is modeled as two fixed actuators [117], i.e. $n_p = n_{p_f} + n_{n_r} = n_{p_f} + 2n_{n_r}$. In the proposed approach the ship is an over-actuated system, because of the number of actuator $n_p > q$, where q is the number of the degree of freedom of

the vessel. The actuators' position vector is defined as

$$r = \begin{bmatrix} l_x & l_y \end{bmatrix}' \tag{4.48}$$

with l_x and l_y the vectors of coordinates in the ship body reference system. The generalized force input vector that the actuators can produce is given by

$$\bar{u}_{r_k} = \bar{A} f_{r_k}. \tag{4.49}$$

In Eq. (4.49), $\bar{A} = \bar{A}(r, \alpha) \in \mathbb{R}^{(n_{\bar{u}} \times n_p)}$ is the constant configuration matrix of the ship

$$\bar{A} = \begin{bmatrix} \cos\alpha_1 & \dots & \cos\alpha_p \\ \sin\alpha_1 & \dots & \sin\alpha_p \\ -l_{y_1}\cos\alpha_1 + l_{x_1}\cos\alpha_1 & \dots & -l_{y_p}\cos\alpha_1 + l_{x_1}\cos\alpha_p \end{bmatrix} \tag{4.50}$$

with $\alpha \in \mathbb{R}^{n_p}$ the vector of the constant actuators' orientations, and $f_{r_k} \in \mathbb{R}^{n_p}$ the vector of real forces provided by the actuators set driven by the extended thruster vector $f_{*_k} \in \mathbb{R}^{n_p}$ provided by the TA algorithm:

$$f_{*_k} = \bar{A}^{\dagger} \bar{u}_k. \tag{4.51}$$

In Eq. (4.51), $\bar{A}^{\dagger}$ is the generalized Moore-Penrose right inverse matrix

$$\bar{A}^{\dagger} = \bar{A}' \left(\bar{A}\bar{A}'\right)^{-1} \tag{4.52}$$

and the solution of the allocation $f_{*_k} = \bar{A}^{\dagger}\bar{u}_k$ is the weighed 2-norm sense optimal solution for an identity weight matrix $W = I$ of appropriate dimension [118]:

$$f'_{*_k} W f_{*_k} \leq f'_k W f_k \quad \forall \{ f_k \in \mathbb{R}^{n_p} \mid f_k = \bar{A}^{\dagger}\bar{u}_k \}. \tag{4.53}$$

4.3.2.3.2 Model Predictive Control Due to the good performance of the identification task (see Fig.4.37 and Tab.4.20), the LTI model of Eq. (4.47) has been used to design the MPC. As reported in Tab.4.20, the optimization problem formulated, over the considered prediction and control horizons (see Tab. 4.22), is affected by a prediction fit error $< 1\%$, introduced by the LTI model approximation [119,120].

Considering the control scheme given in Fig.4.38, in the proposed solution a quadratic cost is minimized subject to the linear equality constraints formulated

with respect to the space state model of Eq. (4.47)

$$\min_{\Delta\bar{u}} \quad \sum_{i=1}^{N_p} \|Q_{\bar{z}}(\bar{z}_{k+i|k} - r_k)\|_2^2 + \sum_{j=0}^{N_u-1} \|Q_{\Delta\bar{u}}\Delta\bar{u}_{k+j|k}\|_2^2 \tag{4.54a}$$

$$\text{s.t.} \quad x_{k|k} = x(k), \tag{4.54b}$$

$$\bar{x}_{k+i+1|k} = A\bar{x}_{k+i|k} + B\bar{u}_{k+i|k}, \tag{4.54c}$$

$$\bar{z}_{k+i+1|k} = C\bar{x}_{k+i+1|k}, \tag{4.54d}$$

$$\Delta\bar{u}_{k+N_u+j|k} = 0, \tag{4.54e}$$

$$\bar{u}_{k+i|k} \in \bar{\mathbb{U}}, \tag{4.54f}$$

$$i = 0, 1, \ldots, N_p - 1, \tag{4.54g}$$

$$j = 0, 1, \ldots, N_u - 1 \tag{4.54h}$$

where N_p is the prediction horizon, N_u is the control horizon, $Q_{\Delta\bar{u}}$ and $Q_{\bar{z}}$ are the weights matrices, with $Q_{\Delta\bar{u}}$ invertible, $\bar{x}_{k+i|k}$ denotes the prediction of the variable $\bar{x}$ at time $k+i$ based on the information available at time k, $\Delta\bar{u}_{k+i|k}$ is the vector of the input increments, with $\bar{u}_{k-1|k} = \bar{u}(k-1)$, $r_k = \begin{bmatrix} r_{x_k} & r_{y_k} & r_{\psi_k} \end{bmatrix}'$ is the output reference, $\bar{\mathbb{U}}$ is the polyhedral set of constraints on inputs and $\bar{z}_k$ is the considered plant input. Due to the considered Wave Filter (WF) algorithm, output $\bar{z}_k = \bar{y}_{r_k}$, where $\bar{y}_{r_k}$ is the real pose of the ship and $\bar{z}_k$ the filtered one. The problem (4.54) can be casted into a condensed QP problem of the form

$$\begin{aligned} \min_{\mu} \quad & \frac{1}{2}\mu' H \mu + \rho_k' F' \mu \\ \text{s.t.} \quad & G\mu \leq V\bar{\rho}_k + w \end{aligned} \tag{4.55}$$

where $\mu \in \mathbb{R}^{n_\mu}$ is the vector of optimization variables, $\rho_k \in \mathbb{P}$ and $\bar{\rho}_k \in \bar{\mathbb{P}}$ are vectors of MPC parameters of dimension $n_{\bar{\rho}} = n_{\bar{u}} + n_{\bar{x}} + n_{\bar{z}}$ and $n_\rho = n_{\bar{u}}$, with $\mathbb{P} \subset \mathbb{R}^{n_\rho}$ and $\bar{\mathbb{P}} \subset \mathbb{R}^{n_{\bar{\rho}}}$ bounded sets of interest with $\bar{\rho}_k = \begin{bmatrix} \bar{u}_{k-1} \end{bmatrix}$ and $\rho_k = \begin{bmatrix} \bar{u}_{k-1} & \bar{x}_k & \bar{z}_k \end{bmatrix}'$, $H \in \mathbb{R}^{n_\mu \times n_\mu}$ is a symmetric and positive definite matrix, $F \in \mathbb{R}^{n_\mu \times n_\rho}$ is the linear cost term, $G \in \mathbb{R}^{m_{\bar{u}} \times n_\mu}$, $V \in \mathbb{R}^{m_{\bar{u}} \times n_{\bar{\rho}}}$ and $w \in \mathbb{R}^{m_{\bar{u}}}$ are the terms defining the inputs' constraints.

Remark 1. *The MPC of Eq. (4.37) has been developed with respect to the LTI system of Eq. (4.47) in order to formulate a LTI-MPC. This approach represents a faster [121, 122] and certifiable [56] solution with respect to alternative nonlinear MPC algorithms [123] that can not-guarantee to maintain the convexity of the quadratic minimization problem [4]. The possibility to certify the algorithm complexity and the computational effort required to compute the control law is of paramount importance for a real-time implementation of the algorithm in commercial control platform. For a comparative performance anal-*

ysis between LTI-MPC and nonlinear MPC algorithms, in particular to control the Cybership II vessel, it is possible to refer to [124, 125].

Remark 2. *Due to the formulation of the TA algorithm in the optimization problem of Eq. (4.53), the force allocation could be directly integrated within the MPC controller [126]. In the proposed approach the TA and the MPC have been decoupled for two reasons: i) due to the study of the actuators' configuration, the TA policy can be developed and used to simplify the identification task of Sec. 4.3.2.2 representing the nonlinear plant by a square linear model. When a system is provided with a trust allocation policy, usually the controller is decoupled from the TA, that run with a different execution rate and it is assumed as part of the controlled plant. In this thesis has been considered this approach to identify the LTI model of the nonlinear plant and to develop the proposed controller; ii) maintaining decoupled the TA and the MPC allows to introduce in the controller the FT policy proposed in the next section.*

4.3.2.4 Fault Tolerant Model Predictive Control

The previous QP problem of Eq. (4.55) is the standard MPC formulated with respect to the vessel model of Eq. (4.47). It guarantees control performances in nominal scenario. However in case of actuators fault the control has low effectiveness. In this thesis a direct feedback has been introduced from the TA block to the MPC in order to consider the actuators' fault effect into the control law. Such feedback considers the direct knowledge of the computed actuator force reference value f_* and of the real actuators force f_r provided by the actuator sensors (see Fig.4.38) [127]. Only the control effort constraints have been considered in the minimization problem (4.54f):

$$\bar{u}_{k+i|k} \in \bar{\mathbb{U}}=\{\bar{u}_{k+i|k} \in \mathbb{R}^{n_{\bar{u}}} | G_{\bar{u}} \Delta \bar{u}_{k+i|k} \leq V_{\bar{u}} \bar{u}_{k+i-1|k} + \bar{u}_C\} \tag{4.56}$$

where

$$\begin{aligned} G_{\bar{u}} = \begin{bmatrix} -I \\ I \end{bmatrix} \ , \ V_{\bar{u}} = \begin{bmatrix} I \\ -I \end{bmatrix} \ , \ \bar{u}_C = \begin{bmatrix} -\bar{u}_m \\ \bar{u}_M \end{bmatrix} \\ \mu = \begin{bmatrix} \Delta \bar{u}_{k|k} & \dots & \Delta \bar{u}_{k+N_u-1|k} \end{bmatrix}' \end{aligned} \tag{4.57}$$

with $I \in \mathbb{R}^{n_{\bar{u}} \times n_{\bar{u}}}$ and

$$\begin{aligned} \bar{u}_m = \begin{bmatrix} \tau_{X_m} & \tau_{Y_m} & \tau_{N_m} \end{bmatrix}' \\ \bar{u}_M = \begin{bmatrix} \tau_{X_M} & \tau_{Y_M} & \tau_{N_M} \end{bmatrix}' \end{aligned} \tag{4.58}$$

the vectors of minimum and maximum control effort bounds, respectively. The QP problem constraints' matrices are:

$$G = \begin{bmatrix} G_{\bar{u}} & 0 & \dots & \dots & 0 \\ G_{\bar{u}} & G_{\bar{u}} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ G_{\bar{u}} & \dots & \dots & \dots & G_{\bar{u}} \end{bmatrix}, \; V = \begin{bmatrix} V_{\bar{u}} \\ \vdots \\ V_{\bar{u}} \end{bmatrix}, \; w = \begin{bmatrix} \bar{u}_C \\ \vdots \\ \bar{u}_C \end{bmatrix}. \tag{4.59}$$

In the same way, the hessian matrix H and the linear term matrix $\rho'_k F'$ in the QP problem (4.55) have been defined as

$$\begin{aligned} H &= (\breve{H}'\breve{H} + \bar{Q}_{\Delta\bar{u}}) = (\mathring{H} + \bar{Q}_{\Delta\bar{u}}) \\ F\rho_k &= \breve{H}\bar{Q}_{\bar{z}}(\mathring{F}x_k - S_k) \end{aligned} \tag{4.60}$$

where $\breve{H} = \breve{H}_{(A,B,C)} \in \mathbb{R}^{(n_{\bar{y}} \cdot N_p) \times (n_{\bar{u}} \cdot N_p - N_c)}$ is named prediction matrix, $\mathring{H} \in \mathbb{R}^{(n_{\bar{u}} \cdot N_p - N_c) \times (n_{\bar{u}} \cdot N_p - N_c)}$, $\mathring{F} = \mathring{F}_{(A,C)} \in \mathbb{R}^{(N_p \cdot n_{\bar{z}}) \times n_{\bar{x}}}$, $\bar{Q}_{\bar{z}} \in \mathbb{R}^{(N_p \cdot n_{\bar{z}}) \times (N_p \cdot n_{\bar{z}})}$ and $\bar{Q}_{\Delta\bar{u}} \in \mathbb{R}^{(N_u \cdot n_{\bar{u}}) \times (N_u \cdot n_{\bar{u}})}$ are weight matrices built from $Q_{n_{\Delta\bar{u}}}$ and $Q_{n_{\bar{z}}}$, $S_k \in \mathbb{R}^{N_p \cdot n_{\bar{z}}}$ such that $S_k = \begin{bmatrix} r_k & \dots & r_k \end{bmatrix}'$ and

$$\breve{H} = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \dots & \dots & \dots & \\ CA^{N_p-1}B & \dots & \dots & CA^{N_p-N_u}B \end{bmatrix}, \; \mathring{F} = \begin{bmatrix} CA \\ CA^2 \\ \dots \\ CA^{N_p} \end{bmatrix}. \tag{4.61}$$

The input constraints (4.54f) formulated by the Eq. (4.61) guarantee to compute a nominal control effort $\bar{u}_k$ satisfying the input signals bounds of Eq. (4.45) considered in the previous identification task of Sec. 4.3.2.2. As discussed before, the provided control effort is not applied directly to the system. This is due to the presence of the TA algorithm that allows to compute the optimal extended force vector f_{*_k} with respect to the nominal matrix $\bar{A}^\dagger$. The real applied control effort is then computed as $\bar{u}_{r_k} = \bar{A} f_k$, by the Moore-Penrose pseudo-inverse proprieties $\bar{A}\bar{A}^\dagger = I$. Considering an actuators' fault scenario, $\bar{u}_{r_k} \neq \bar{u}_k$ because of the computed optimal extended force vector $f_{*_k} \neq f_{r_k}$, where f_{r_k} is the real extended force vector provided by the actuators.

Remark 3. *Control effort estimation is a widely studied problem and several solutions have been proposed in order to provide an estimation useful for fault detection and tolerance algorithms [128–132]. These results allow to consider the estimated value instead of the real force provided by each thruster.*

If $\bar{A}^\dagger$ provides the thrust allocation result in nominal conditions and $\bar{u}_{r_{k-1}} =$

$\bar{A}f_{r_{k-1}}$, defining the diagonal matrix $\bar{f}_k \in \mathbb{R}^{(n_p \times n_p)}$

$$\bar{f}_k = \begin{bmatrix} \left(\frac{f^{(1)}_{r_{k-1}}}{f^{(1)}_{*_{k-1}}}\right) & 0 & \dots & 0 \\ 0 & \dots & 0 & \left(\frac{f^{(n_p)}_{r_{k-1}}}{f^{(n_p)}_{*_{k-1}}}\right) \end{bmatrix} \tag{4.62}$$

where $f^{(i)}_{*_{k-1}}$ is the nominal force required by the $i-$th actuator and $f^{(i)}_{r_{k-1}}$ is the real one provided at previous time instance, for $i = 1, \dots, n_p$, the mapping matrix in case of fault is $\bar{A}_k = \bar{A}\bar{f}_k$ such that

$$\bar{u}_{r_k} = \begin{cases} \bar{A}_k \bar{A}^\dagger \bar{u}_k & \text{if } f_{r_k} \neq f_{*_k} \\ \bar{u}_k, & \text{otherwise.} \end{cases} \tag{4.63}$$

The equality $f_{r_{k-1}} = f_{*_{k-1}}$, tested at each time step k successive to the application of the control effort $f_{r_{k-1}}$, is the fault detection condition. In fault scenarios, the plant (4.47) could not fit the real system behavior due to unavailable actuators and to the relative reduction of system governability. From Eq. (4.63), the fault matrix $M_k \in \mathbb{R}^{n_{\bar{u}} \times n_{\bar{u}}}$ is defined and iteratively computed

$$M_k = \bar{A}_k \bar{A}^\dagger = \bar{A}\bar{f}_k \bar{A}^\dagger \tag{4.64}$$

in order to introduce the fault effect in the plant model $\mathcal{S}_{f_k}$:

$$\begin{aligned} \bar{x}(k+1) &= A\bar{x}(k) + BM_k\bar{u}(k) \\ \bar{z}(k) &= C\bar{x}(k) \end{aligned} \tag{4.65}$$

where $M_k = I$ in absence of fault. The definition of $\mathcal{S}_{f_k}$ of Eq. (4.65) suggest to rebuild the MPC with respect to the new system in order to formulate the fault tolerant policy.

4.3.2.4.1 Fault Tolerant Algorithm The introduction of M_k into the plant model of Eq. (4.65) causes two drawbacks:

- replacing in the model plant (4.47) the input matrix B with BM_k causes the lost of knowledge of the real plant input $\bar{u}_r$, which in absence of fault has been constrained as proposed in Eq. (4.63), and in fault scenario cannot be constrained because the control law provides only the TA input $\bar{u}_k$, losing the knowledge about $\bar{u}_{r_k}$;
- $\mathcal{S}_{f_k}$ of Eq. (4.65) is a Linear Time-Varying (LTV) system and standard LTV-MPC requires to rebuild and also to solve the QP problem iteratively, introducing a strong computational effort in the algorithm execution [96].

In order to overcome the above problems, it is possible to introduce directly the fault effect into the QP problem of Eq. (4.55) defining the block diagonal matrix

$$\bar{M}_k = \begin{bmatrix} M_k & 0 & \dots & 0 \\ 0 & M_k & \dots & 0 \\ 0 & \dots & 0 & M_k \end{bmatrix} \tag{4.66}$$

where $\bar{M}_k \in \mathbb{R}^{(n_u \cdot N_u)\times(n_u \cdot N_u)}$ and solving iteratively the problem

$$\begin{aligned} \min_{\mu} \quad & \frac{1}{2}\mu' \bar{H}_k \mu + \rho_k' \bar{F}_k \mu \\ \text{s.t.} \quad & \bar{G}_k \mu \leq \bar{V}_k \bar{\rho}_k + w. \end{aligned} \tag{4.67}$$

The problem matrices are

$$\begin{aligned} \bar{H}_k &= \bar{M}_k' \mathring{H} \bar{M}_k + \bar{Q}_{\Delta\bar{u}} \\ \bar{F}_k &= F' \bar{M}_k \\ \bar{G}_k &= G \bar{M}_k \\ \bar{V}_k &= V \bar{M}_k \end{aligned} \tag{4.68}$$

with $\bar{H}_k \in \mathbb{R}^{n_\mu \times n_\mu}$, $\bar{F}_k \in \mathbb{R}^{n_\mu \times n_\rho}$, $\bar{G}_k \in \mathbb{R}^{m_{\bar{u}} \times n_\mu}$, $\bar{V}_k \in \mathbb{R}^{m_{\bar{u}} \times n_\rho}$. In the case of not actuator fault, by Eq. (4.63) $\bar{H}_k = H$, $\bar{F}_k = F$, $\bar{G}_k = G$, $\bar{V}_k = V$.

Both MPC and FT-MPC algorithms requires the estimated state $\bar{x}_k$ in order to compute iteratively the control law and in standard MPC it is usually provided by a Kalman Filter (KF). In fault scenarios, also the nominal KF designed with respect to nominal plant $\mathcal{S}$ of Eq. (4.47) need to be re-formulated. In particular, in this thesis it has been done replacing in the KF algorithm the nominal B matrix with the recomputed one BM_k referred to $\mathcal{S}_k$ of Eq. (4.65).

4.3.2.4.2 Non Controllable Scenarios The proposed approach allows to control the plant in the presence of actuators' fault and also to rebuild correctly the input constraints with respect to the real applied input avoiding the above considered drawbacks. Furthermore, by the presented solution it is possible to study the conditions of plant controllability in case of fault. Under the assumption that the nominal LTI plant model $\mathcal{S}$ of Eq. (4.47) is controllable

$$\text{rank}\left(\underbrace{\begin{bmatrix} B & AB & \dots & A^{n_{\bar{x}}-1}B \end{bmatrix}}_{R_{(A,B)}} \right) = n_{\bar{x}} \tag{4.69}$$

the controllability of the plant in case of fault, i.e. $\mathcal{S}_{f_k}$ of Eq. (4.65), is given by $\text{rank}(R_{(A,BM_k)}) = n_{\bar{x}}$. This condition can be verified iteratively in case

of fault in order to check if it is allowed to maintain the control of the vessel or otherwise to set to a safety state the control action. Furthermore the above controllability condition can be used to define offline the not controllable scenarios set according to the value of each element of $\bar{f}_k$ testing for which combination of values $0 \leq f_{r_k}^{(i)}/f_{*_k}^{(i)} \leq 1$ with $i = 0, \ldots, n_p$, the system becomes not controllable, with $f_{*_k}^{(i)} \neq 0$.

To identify not controllable fault scenarios it is necessary to verify only the combinations of fault conditions causing a full breakdown (failure) of some actuator $f_{r_k}^{(i)}/f_{*_k}^{(i)} = 0$ and then to check if $\bar{f}_k \in \mathbb{F}$ where

$$\mathbb{F} = \left\{ \bar{f} \in \mathbb{R}^{n_p} | \text{rank}\left(R_{(A,BM_k)}\right) < n_{\bar{x}} \right\} \tag{4.70}$$

is the not controllable actuators' fault set, with $\mathbb{F} \subset \mathbb{R}^{\bar{f}}$ where $\mathbb{R}^{\bar{f}}$ is the actuators fault subspace $\mathbb{R}^{\bar{f}} \subset \mathbb{R}^{n_p}$ defined as

$$\mathbb{R}^{\bar{f}} = \{ \bar{f} \in \mathbb{R}^{n_p} \mid 0 \leq \bar{f}^{(i)} \leq \bar{f}_*^{(i)} \}. \tag{4.71}$$

The tested Fault Tolerant MPC algorithm is reported in Algorithm 1. The system is stopped in the presence of an actuators' fault verifying the not controllability condition.

4.3.2.5 Simulation results

In this section simulation results comparing standard MPC and the proposed FT-MPC have been reported. The not controllable set of actuators' faults has been studied, considering the unavailability of actuators affected by faults by setting $f_r^{(i)} = 0$ for each i-*th* lost actuator and $f_r^{(i)} = f_*^{(i)}$ for the other ones. The not controllable set $\mathbb{F}$ obtained by the study of the condition of the Eq. (4.70) has been shown in Tab. 4.21, reporting the actuators fault combinations driving the system in the safety state such that $u_k = [0, 0, 0]'$. Due to the actuators' configuration, fault scenarios setting the model into a not controllable state are related to a controllability matrix $R_{(A,BM_k)}$ featured by a rank value smaller than the size of the considered space-state representation of Eq. (4.47) with $n_{\bar{x}} = 10$.

In order to test the effectiveness of the considered algorithm a set of fault scenarios have been considered, driving the vessel by a set of test trajectories in case of several actuators' faults. The considered MPC tuning parameters have been reported in Tab. 4.22. In the proposed simulations the additive noise $d(k)$ has been generated by the model proposed in Eq. (4.46). The MATLAB active-set based QP solver quadprog is used in the reported simulations. Both MPC and FT-MPC have been tested driving the system with a set of step reference signals of amplitude $r_x = 10$ m, $r_y = 5$ m and $r_\psi = 0.0175$ rad, introducing

Table 4.21: Not controllable actuators' fault combinations

Fault scenario					
$f_r^{(1)}$	$f_r^{(2)}$	$f_r^{(3)}$	$f_r^{(4)}$	$f_r^{(5)}$	$\operatorname{rank}\left(R_{(A,BM_k)}\right)$
$f_*^{(1)}$	0	0	0	0	7
0	$f_*^{(2)}$	0	0	0	8
0	0	$f_*^{(3)}$	0	0	7
0	0	0	$f_*^{(4)}$	0	7
0	0	0	$f_*^{(4)}$	$f_*^{(5)}$	7
0	0	0	0	$f_*^{(5)}$	7
0	0	0	0	0	0

Table 4.22: TA and MPC Tuning Parameters and Constraints

Parameter	Symbol	Value
Prediction horizon	N_p	15
Control horizon	N_u	5
MO weights	$Q_{\bar{z}}$	$\begin{bmatrix}1 & 1 & 2\end{bmatrix}$
MV rate weights	$Q_{\Delta\bar{u}}$	$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 5\end{bmatrix}$
MV constraints		$-2 \leq \tau_X \leq 2$
		$-1.5 \leq \tau_Y \leq 1.5$
		$-1.5 \leq \tau_N \leq 1.5$
Sampling time	T_s	0.25 s

Algorithm 1 FT-MPC Algorithm

Input: Previous control input $\bar{u}_{k-1}$, estimated state $\bar{x}_k$, plant output $\bar{z}_k$, previous nominal forces vector $f_{*_{k-1}}$, previous real forces vector $f_{r_{k-1}}$, mapping matrix $\bar{A}$, inverse mapping matrix $\bar{A}^\dagger$, not controllability set $\mathbb{F}$, previous M_{k-1}, QP matrices H, F, G, V, w, $\mathring{H}$, $\bar{Q}_{\Delta n_{\bar{u}}}$

$\rho_k \leftarrow \begin{bmatrix}\bar{u}_{k-1} & \bar{x}_k & \bar{z}_k\end{bmatrix}'$;
$\bar{\rho}_k \leftarrow \bar{u}_{k-1}$;
SafetyState $\leftarrow 0$;
$\bar{f}_k \leftarrow (f_{*_{k-1}}, f_{r_{k-1}})$; ▷ See Eq. (4.62)
if $\bar{f}_k \in \mathbb{F}$ **then**
    SafetyState $\leftarrow 1$;
    $M_k \leftarrow \bar{A}\bar{f}_k\bar{A}^\dagger$;
    $\bar{M}_k \leftarrow M_k$; ▷ See Eq. (4.66)
    $\bar{H}_k \leftarrow \bar{M}_k'\mathring{H}\bar{M}_k + \bar{Q}_{\Delta n_{\bar{u}}}$;
    $\bar{F}_k \leftarrow F'\bar{M}_k$;
    $\bar{G}_k \leftarrow G\bar{M}_k$;
    $\bar{V}_k \leftarrow V\bar{M}_k$;
    $\bar{H}_k \leftarrow H$; $\bar{F}_k \leftarrow F'$; $\bar{G}_k \leftarrow G$; $\bar{V}_k \leftarrow V$;$M_k \leftarrow I$;
end if
if SafetyState $= 0$ **then**
    $\mu^* \leftarrow \min_{\mu} \frac{1}{2}\mu'\bar{H}_k\mu + \rho_k'\bar{F}_k\mu$ s.t. $\bar{G}_k\mu \leq \bar{V}_k\bar{\rho}_k + w$;
    $\Delta\bar{u}_k \leftarrow \mu^*$;
    $\bar{u}_k \leftarrow \bar{u}_{k-1} + \Delta\bar{u}_k$;
    $\bar{u}_k \leftarrow [0, 0, 0]'$;
end if

Output: The inputs $\bar{u}_k$ to be applied to the system.

a wave disturbance featured by a noise intensity $\sigma^i = 0.5$. The reference signal for the ship orientation r_ψ has been filtered by an appropriate low-pass filter introduced in order to provide prescribed smooth output behavior. A simulation time of 100 s has been considered in each actuators' fault scenario.

The MPC and FT-MPC algorithms have been compared with respect to the percentage Integral Square Error (ISE) improvement:

$$\Delta\text{ISE} = \frac{\text{ISE}_{MPC} - \text{ISE}_{FT-MPC}}{\text{ISE}_{MPC}} \times 100. \tag{4.72}$$

In Tab. 4.23 the ΔISE results related to the considered step response for each controlled output have been reported with the pedics x, y and ψ for each controlled output. The first row of the Tab. 4.23 presents the ΔISE value in case of normal behavior, i.e. in absence of actuators' fault, and the other rows compare the MPC and FT-MPC algorithms for each faults' combination. The average percentage ISE improvements of the FT-MPC with respect to the MPC are

13.47%, 25.47% and 95.21% for x, y and ψ, respectively. In particular, the proposed fault tolerant policy allows to eliminate the transient overshoot related to the controlled orientation ψ and to reduce the controlled output steady-state error variance (5% with the FT-MPC instead of the 8% of the reference value with the standard MPC). These results validate the effectiveness of the proposed approach, showing the improvement of the FT-MPC with respect to the standard MPC in each fault scenarios, and maintaining standard performance in case of fault free behavior. The reported ISE results show the reduction of the error obtained applying the FT-MPC in particular with respect to the orientation error which in case of fault is the most affected by performance degradation.

As an example, in Figs. 4.40-4.43 simulations related to one of the proposed actuators' fault scenarios, i.e. considering $f_r = \begin{bmatrix} f_*^{(1)} & 0 & f_*^{(3)} & 0 & f_*^{(5)} \end{bmatrix}'$, have been reported, also varying the wave intensity parameter in order to show the transient response improvement given by the introduction of the fault tolerant action in the standard MPC. In the simulations of Figs.4.40-4.43 the system has been governed with two consecutive equal steps. During the first step the fault-free scenario has been considered (see time instant $T < 100$) and in the second (see time instant $T \geq 100$) the considered actuators' faults configuration has been introduced in the vessel. The results show simulation considering an additive wave noise featured by an intensity of $\sigma^i = 0.5$, equivalent to the noise considered in the identification task. In particular in Fig. 4.40 the compared control results have been reported, showing the improvement introduced by the proposed approach with respect to the standard MPC controlling each driven output in a fault scenario. Fig. 4.41 shows the control efforts comparison, showing that the introduction of the fault effects in the reference plant model causes the computation of different control signals, allowing to improve the control performances. The compared algorithms provide the same control results in the first part of the simulation of Figs. 4.40-4.41, i.e. when the nominal fault-free scenario has been considered.

Furthermore, considering the previous fault scenario, simulation results considering a wave noise featured by an intensity of $\sigma^i = 5$ have been reported, in order to test the control performance in case of strong wave additive disturbance. Fig. 4.42 and Fig. 4.43 shown the controlled outputs and the control efforts in fault-free and in faulty scenario. In fault-free control scenario ($T < 100$) the compared controllers provide the same results. For $T \geq 100$ the fault is introduce in the ship, showing that the FT-MPC provides again good control performances ($\Delta\text{ISE}_x = 15.42\%$, $\Delta\text{ISE}_y = 8.04\%$, $\Delta\text{ISE}_\psi = 87.35\%$). The results of the FT-MPC are similar to that obtained in previous case of low wave noise while the control results provided by the standard MPC show lower performance. Also in these conditions of strong disturbance the orientation ψ

Table 4.23: Actuators' fault scenarios control results

Fault scenario					Δ ISE$_x$	Δ ISE$_y$	Δ ISE$_\psi$
$f_r^{(1)}$	$f_r^{(2)}$	$f_r^{(3)}$	$f_r^{(4)}$	$f_r^{(5)}$	[%]	[%]	[%]
$f_*^{(1)}$	$f_*^{(2)}$	$f_*^{(3)}$	$f_*^{(4)}$	$f_*^{(5)}$	0	0	0
0	0	$f_*^{(3)}$	0	$f_*^{(5)}$	4.65	17.95	98.78
0	0	$f_*^{(3)}$	$f_*^{(4)}$	$f_*^{(5)}$	−1.00	0.32	21.03
0	0	$f_*^{(3)}$	$f_*^{(4)}$	0	4.65	17.95	98.76
0	$f_*^{(2)}$	0	0	$f_*^{(5)}$	27.37	25.19	99.90
0	$f_*^{(2)}$	0	$f_*^{(4)}$	$f_*^{(5)}$	26.82	21.56	99.90
0	$f_*^{(2)}$	0	$f_*^{(4)}$	0	27.37	25.19	99.90
0	$f_*^{(2)}$	$f_*^{(3)}$	0	0	20.27	8.38	99.83
0	$f_*^{(2)}$	$f_*^{(3)}$	0	$f_*^{(5)}$	21.04	12.03	99.05
0	$f_*^{(2)}$	$f_*^{(3)}$	$f_*^{(4)}$	$f_*^{(5)}$	20.94	1.43	96.40
0	$f_*^{(2)}$	$f_*^{(3)}$	$f_*^{(4)}$	0	21.04	12.030	99.05
$f_*^{(1)}$	0	0	0	$f_*^{(5)}$	19.90	12.05	99.78
$f_*^{(1)}$	0	0	$f_*^{(4)}$	$f_*^{(5)}$	19.52	19.07	99.88
$f_*^{(1)}$	0	0	$f_*^{(4)}$	0	19.90	12.50	99.78
$f_*^{(1)}$	0	$f_*^{(3)}$	0	0	20.47	1.25	99.82
$f_*^{(1)}$	0	$f_*^{(3)}$	$f_*^{(4)}$	$f_*^{(5)}$	18.45	−0.91	93.80
$f_*^{(1)}$	0	$f_*^{(3)}$	$f_*^{(4)}$	0	21.62	11.03	99.72
$f_*^{(1)}$	$f_*^{(2)}$	0	0	0	−0.07	−0.05	99.57
$f_*^{(1)}$	$f_*^{(2)}$	0	$f_*^{(4)}$	$f_*^{(5)}$	0.63	25.87	99.92
$f_*^{(1)}$	$f_*^{(2)}$	0	$f_*^{(4)}$	0	0.58	24.93	99.89
$f_*^{(1)}$	$f_*^{(2)}$	$f_*^{(3)}$	0	0	−0.02	−5.64	99.87
$f_*^{(1)}$	$f_*^{(2)}$	$f_*^{(3)}$	0	$f_*^{(5)}$	1.16	3.43	99.64
$f_*^{(1)}$	$f_*^{(2)}$	$f_*^{(3)}$	$f_*^{(4)}$	0	1.16	3.43	99.64

presents a good improvement due to the introduction of the fault tolerant action, providing results comparable with those obtained in the fault-free scenario in terms of transient overshoot compensation and steady-state stabilization error variance (10% with the FT-MPC instead of the 25% of the reference value with the standard MPC).

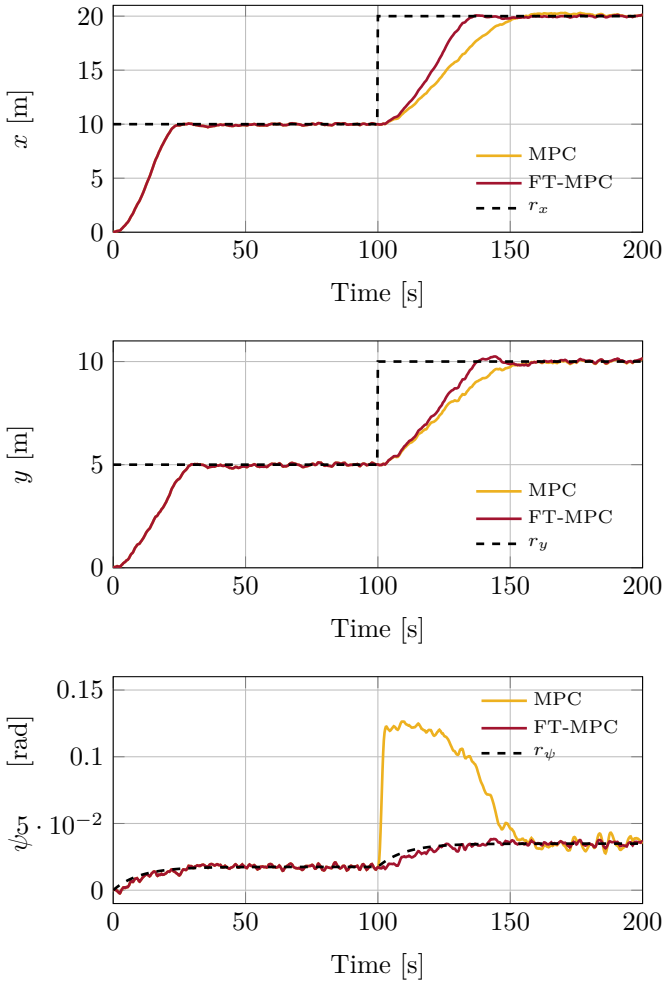

Figure 4.40: Comparison of the output position and asset controlling the Cybership II with the MPC (yellow) and the FT-MPC (red) algorithms.

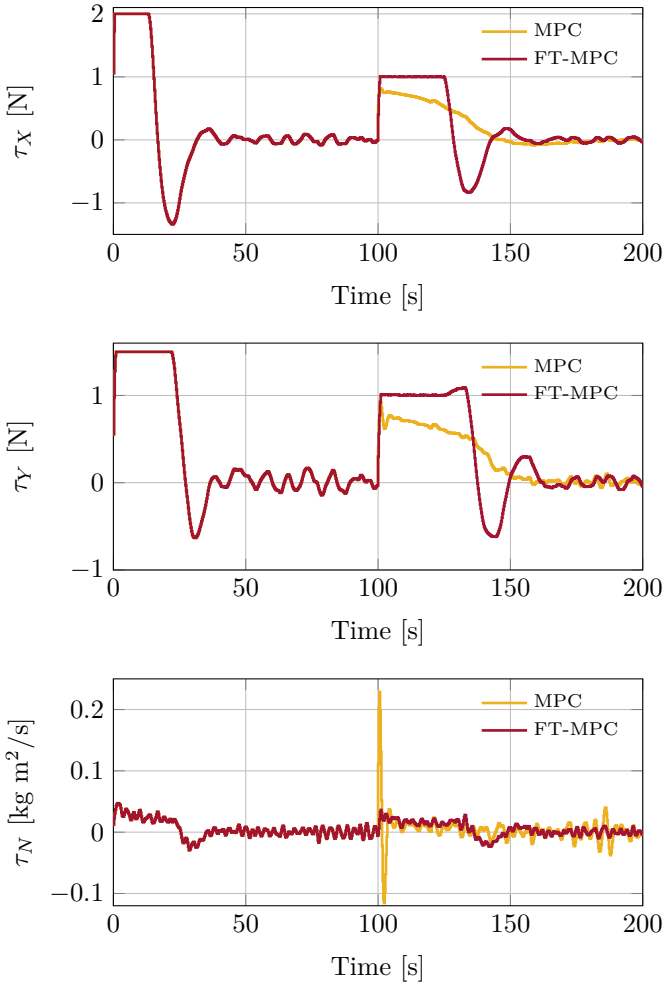

Figure 4.41: Comparison of the control efforts controlling the Cybership II with the MPC (yellow) and the FT-MPC (red) algorithms.

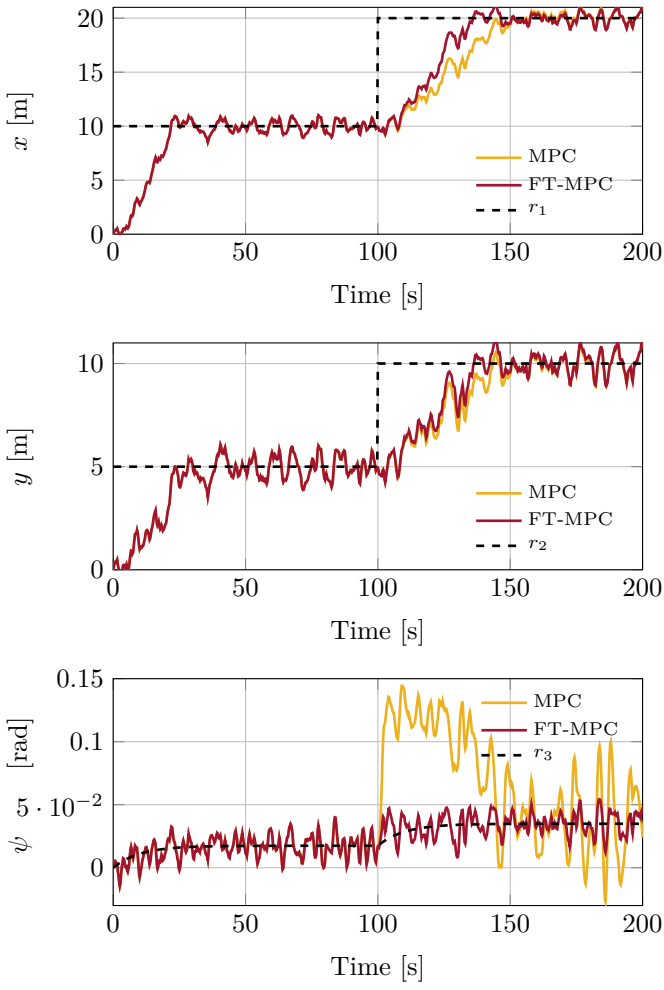

Figure 4.42: Comparison of the output position and asset controlling the Cybership II with the MPC (yellow) and the FT-MPC (red) algorithms in the considered scenario.

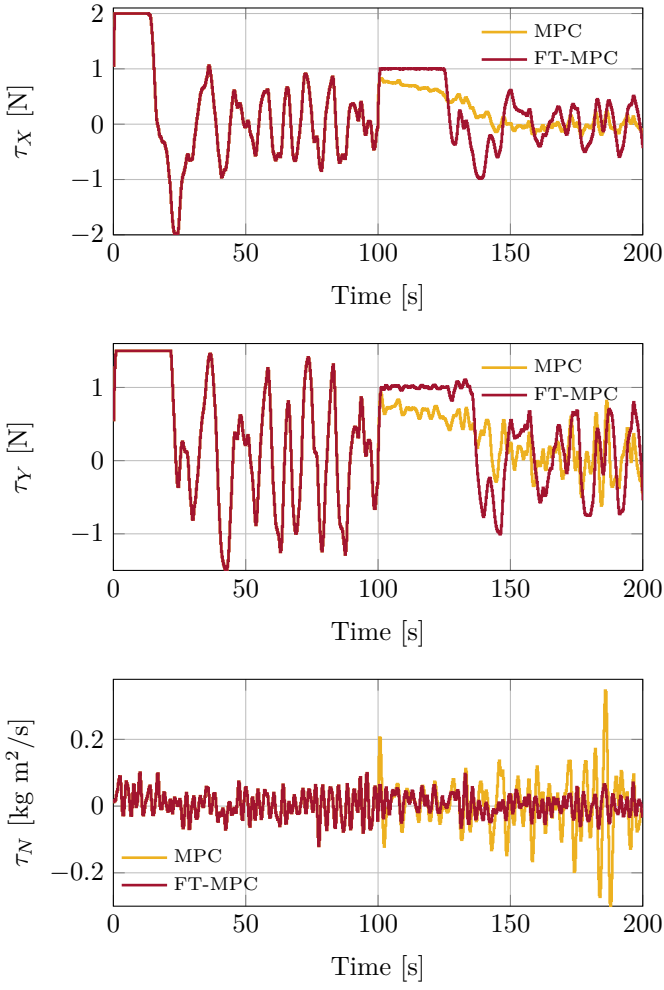

Figure 4.43: Comparison of the control efforts controlling the Cybership II with the MPC (yellow) and the FT-MPC (red) algorithms in the considered scenario.

Chapter 5

MPC for LPV systems

In this chapter the MPC algorithms for Linear Parameter-Varying (LPV) models are presented. The LPV paradigm, presented in Sec.3.1, represents a standard approach to model nonlinear dynamics systems considering the nonlinearities grouped in a set of exogenous estimated or measured time-varying parameters. By the iterative update of the parameters' value, the LPV model allows to describe the nonlinear dynamics through a piece-wise space-state formulation closed to the Linear Time Invariant (LTI) system form. LPV-MPC exploits the LPV paradigm to iteratively reformulate the controller with respect to the instantaneous value of the parameters. This approach presents several problems related to the reformulation of the control law [133].

This chapter is organized as follows. Sec. 5.1 presents the LPV-MPC policy proposed in this thesis to control the nonlinear model of an aircraft, named Aerosonde [134]. The nonlinear dynamics of the aircraft, approximated by a LPV model, has been controlled by a MPC featured by the anticipative action presented in Sec. 5.1.3. In order to improve the control performances, an estimation of the future scheduling parameters value has been introduced in the control law formulation. Furthermore, a MPC policy to drive the aircraft altitude by an external control loop has been proposed and compared with a switching MPC technique. Section 5.2 presents an innovative algorithm to reduce the computational time required by standard LPV-MPC. In the proposed algorithm the LPV-MPC control law re-computation step can be avoided by the introduction of a coordinate transformation, related to the instantaneous parameter array value. The proposed solution has been tested on a benchmark test to validate the sub-optimality of the considered algorithm. Furthermore, a real world case study has been considered, testing the solution with respect to the standard LPV-MPC in a catenary-free tram model.

5.1 LPV-MPC for Aerosonde Aircraft

In this section two techniques based on previewing [135], also called anticipative action, have been proposed in order to improve the performances of the MPC-

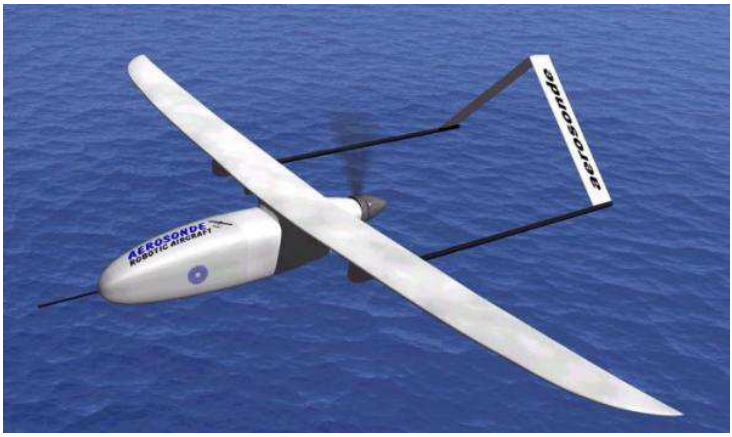

Figure 5.1: The Aerosonde UAV

based autopilot for the Aerosonde [134]. The UAV is modelled in the form of a LPV system by the analysis of the simulated aircraft data performed by the Predictor Based Subspace IDentification (PBSID) toolbox [102]. A Previewing MPC_{LPV} ($\text{P-MPC}_{\text{LPV}}$) [136] based on the identified model is developed in order to control with this inner loop the aircraft asset. Due to the linearization effect of the inner loop, a Previewing MPC_{LTI} ($\text{P-MPC}_{\text{LTI}}$ based on the identified LTI model of the controlled asset system) is designed to drive the UAV altitude system by a secondary slower control loop. The technique proposed for the internal $\text{P-MPC}_{\text{LPV}}$ controller considers the last computed optimization sequence to estimate the future time-varying parameters behaviour improving the control results. Furthermore, considering the development of an ad-hoc controller for each fly scenario, a technique that switch between different $\text{P-MPC}_{\text{LTI}}$ for the external altitude control loop has been considered. The proposed algorithms are based on the anticipative action approach, which considers the knowledge in advance of the full reference signal behaviour.

This part of the thesis is organized as follows. In Sec. 5.1.1 the Aerosonde dynamics model is given and it is identified. Sec. 5.1.2 reports the proposed autopilot architecture. In sections 5.1.3 and 5.1.4 the proposed asset control algorithms have been presented. Sections 5.1.5 and 5.1.6 report the altitude control algorithms. In Sec. 5.1.7 and 5.1.8 have been compared the proposed algorithms.

5.1.1 Identification of the Aerosonde LPV model

The Aerosonde is a small UAV designed to collect and store weather data, such as humidity, temperature, pressure and wind speed, over remote and danger-

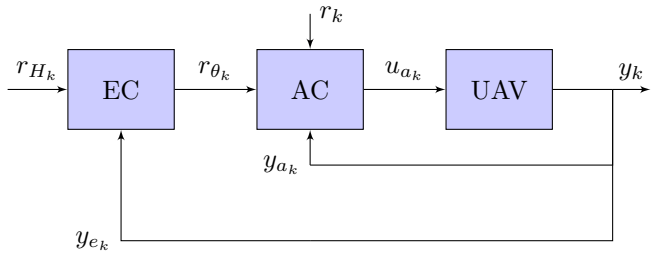

Figure 5.2: Block scheme for the two layer Model Predictive Control Autopilot

Table 5.1: Aerosonde nonlinear model time-varying parameters bounds and fixed input values

δ_m		13	
δ_i		1	
δ_f		0	rad
$V_{a_{max}}$	airspeed upper bound	50	m/s
$V_{a_{min}}$	airspeed lower bound	15	m/s
α_{max}	AOA upper bound	0.3	rad
α_{min}	AOA lower bound	-0.1	rad
β_{max}	sideslip angle upper bound	0.5	rad
β_{min}	sideslip angle lower bound	-0.5	rad
ϕ_{max}	bank angle upper bound	0.5	rad
ϕ_{min}	bank angle lower bound	-0.5	rad
θ_{max}	pitch angle upper bound	0.5	rad
θ_{min}	pitch angle lower bound	-0.25	rad
ψ_{max}	heading angle upper bound	0.75	rad
ψ_{min}	heading angle lower bound	-0.75	rad

ous areas [137]. The vehicle has been developed by Aerosonde Pty Ltd and Aerosonde North America [134] and the UAV nonlinear model is simulated by the Matlab® Aerosim blockset provided by u-dynamics [99]. The dynamical nonlinear behaviour provided by the blockset is the model with six degree of freedom proposed in [138]; complete specifications of the Aerosonde plant and mathematical model are reported in [139]. The UAV model setup considers a set of time-varying parameters and control input bounds as reported in Tab. 5.1. Control inputs related to mixture δ_m, ignition δ_i and lift δ_l are considered fixed in this thesis, studying the aircraft in steady-state fly conditions. A LPV representation of the Aerosonde nonlinear plant has been identified in order to develop the UAV autopilot. In the identification procedure, the nonlinear plant has been simulated with a sampling time $T_s = 4$ ms and the plant data (input, output and parameters) have been acquired with a sampling time

of $T_a = 400$ ms. In the identification task the PBSID toolbox [102] is used and the Aerosonde nonlinear plant is represented as:

$$\begin{aligned} x_{a_{k+1}} &= f(x_{a_k}, u_{a_k}, \rho_k) \\ y_{a_k} &= g(x_{a_k}, u_{a_k}, \rho_k) \end{aligned} \tag{5.1}$$

where $y_{a_k} \in \mathbb{R}^{n_{y_a}}$ is the output vector, $x_{a_k} \in \mathbb{R}^{n_{x_a}}$ is the state vector, $u_{a_k} \in \mathbb{R}^{n_{u_a}}$ is the input vector and $\rho_k \in \mathbb{R}^{n_\rho}$ is the parameters vector of time-varying parameters, unknown a priori but measurable or estimable. This aspect distinguishes the considered LPV systems from linear parameter-varying systems for which the model time-variations are known beforehand. The nonlinear model of the Aerosonde is considered in steady-state fly condition, allowing to study separately longitudinal and latitudinal plant subsystems [140]. By the analysis of the simulated data, the model of Eq. (5.1) is converted into the following LPV system in the form

$$\begin{aligned} x_{a_{k+1}} &= A_k x_{a_k} + B_k u_{a_k} \\ y_{a_k} &= C_k x_{a_k} \end{aligned} \tag{5.2}$$

with $A_k \in \mathbb{R}^{n_{x_a} \times n_{x_a}}$, $B_k \in \mathbb{R}^{n_{x_a} \times n_{u_a}}$ and $C_k \in \mathbb{R}^{n_{y_a} \times n_{x_a}}$ given by

$$A_k = \bar{A}_0 + \sum_{i=1}^{n_\rho} \bar{A}_i \rho_{i_k} \quad , \quad B_k = \bar{B}_0 + \sum_{i=1}^{n_\rho} \bar{B}_i \rho_{i_k} \quad , \quad C_k = \bar{C}_0 + \sum_{i=1}^{n_\rho} \bar{C}_i \rho_{i_k} \tag{5.3}$$

where $\bar{A}_i \in \mathbb{R}^{n_{x_a} \times n_{x_a}}$, $\bar{B}_i \in \mathbb{R}^{n_{x_a} \times n_{u_a}}$, $\bar{C}_i \in \mathbb{R}^{n_{y_a} \times n_{x_a}}$ with $i = 0, \dots n_\rho$ are constant estimated matrices. The input vector is $u_{a_k} = \begin{bmatrix} \delta_{t_k} & \delta_{e_k} & \delta_{a_k} & \delta_{r_k} \end{bmatrix}'$, the output vector is $y_{a_k} = \begin{bmatrix} \theta_k & \phi_k & \psi_k & V_{a_k} \end{bmatrix}'$ and the scheduling parameters vector is $\rho_k = \begin{bmatrix} \alpha_k & \beta_k & M_k & p_k \end{bmatrix}'$. The terms α_k, β_k are the scheduling parameters for the identified LPV latitudinal subsystem of the aircraft and M_k, p_k the scheduling parameters for the longitudinal subsystem. LPV model validation results are grouped in Tab. 5.2. The identified LPV system is compared with respect to a validation set of simulated nonlinear plant data and evaluated varying the prediction horizon N_p in terms of the Variance Accounted For (VAF) value, $\text{VAF} = 100 \cdot \max \left[1 - \frac{var(y_a - \hat{y}_a)}{var(y_a)} \right]$.

5.1.2 Autopilot Structure

The proposed autopilot is structured on a two layer control loops with an asset inner P-MPC$_{\text{LPV}}$ loop (sample time T_a) called Asset Controller (AC) and, due to the linearization effect of the AC, a slower (sample time $T_e = 4$ s) external altitude P-MPC$_{\text{LTI}}$ loop controller called External Controller (EC). The au-

Table 5.2: Identified model Variance Accounted For validation results

	V_a	ϕ	θ	ψ	N_p
VAF	99.3281	98.1254	92.4321	95.3723	3
	98.5046	96.5238	90.8820	93.9795	6
	98.3386	96.0970	89.1098	91.2434	9
	96.3639	94.9137	85.1730	90.4980	12
	92.9994	93.9529	81.3971	88.1975	15
	93.9176	94.7903	77.7614	83.0647	18

topilot control block diagram is shown in Fig. 5.2, where $y_k = \begin{bmatrix} y_{a_k} & y_{e_k} \end{bmatrix}'$ is the plant output vector $y_k \in \mathbb{R}^{(n_{y_a}+n_{y_e})}$, $h_k = y_{e_k} \in \mathbb{R}$ is the external loop controlled output, $r_{a_k} = \begin{bmatrix} r_{\theta_k} & r_k \end{bmatrix}'$ is the asset reference vector ($r_{a_k} \in \mathbb{R}^{n_{y_a}}$), $r_k = \begin{bmatrix} r_{\phi_k} & r_{\psi_k} & r_{V_{a_k}} \end{bmatrix}'$ is the external reference vector for airspeed, heading and bank angle, $r_{\theta_k} \in \mathbb{R}$ if the reference signal for the pitch angle provided by the EC and $r_{H_k} \in \mathbb{R}$ is the reference for the controlled altitude.

5.1.3 Previewing Linear Parameter-Varying MPC for Asset Control

Considering the identified LPV plant of Eq. (5.2) a P-MPC$_{\text{LPV}}$ for the UAV Asset Control (AC) is proposed. P-MPC$_{\text{LPV}}$ solves iteratively a finite-horizon, optimal control problem formulated with respect to a prediction model of the controlled process and an estimation of its current state. When the cost function is quadratic and the constraints are affine, considering the anticipative action, the P-MPC$_{\text{LPV}}$ optimization problem to be solved at each time step can

be formulated as

$$\min_{\Delta u_a} \sum_{i=1}^{N_{p_a}-1} \|Q_a(y_{a_{k+i|k}} - r_{a_k+i})\|_2^2 + \sum_{l=0}^{N_{u_a}-1} \|R_a \Delta u_{a_{k+l|k}}\|_2^2 + \|P_a(y_{a_{k+N_{p_a}|k}} - r_{a_{k+N_{p_a}}})\|_2^2 \tag{5.4a}$$

$$\text{s.t. } x_{a_{k+i+1|k}} = A_k x_{a_{k+i|k}} + B_k u_{a_{k+i|k}} \tag{5.4b}$$

$$y_{a_{k+i+1|k}} = C_k x_{a_{k+i+1|k}} \tag{5.4c}$$

$$x_{a_{k|k}} = x_{a_k} \tag{5.4d}$$

$$\Delta u_{a_{k+N_u+j|k}} = 0 \tag{5.4e}$$

$$r_{a_{k+N_{r_a}+v}} = r_{a_{k+N_{r_a}+v-1}} \tag{5.4f}$$

$$u_{a_{k+l|k}} \in \mathbb{U}_a \tag{5.4g}$$

$$\Delta u_{a_{k+l|k}} \in \mathbb{D}_a \tag{5.4h}$$

$$y_{a_{k+l|k}} \in \mathbb{Y}_a \tag{5.4i}$$

$$i \in \{0, \dots, N_{p_a} - 1\} \tag{5.4j}$$

$$l \in \{0, \dots, N_{u_a} - 1\} \tag{5.4k}$$

$$j \in \{0, \dots, N_{p_a} - N_{u_a} - 1\} \tag{5.4l}$$

$$v \in \{0, \dots, N_{p_a} - N_{r_a} - 1\} \tag{5.4m}$$

where N_{p_a} is the prediction horizon, N_{u_a} is the control horizon, N_{r_a} is the previewing horizon, Q_a, R_a, P_a are weight matrices of appropriate dimension, $x_{a_{k+i|k}}$ denotes the prediction of the variable x_a at time $k+i$ based on the information available at time k, $\Delta u_{a_{k+i|k}}$ is the vector of the input increments, with $u_{a_{k-1|k}} = u_{a_{k-1}}$, r_{a_k+i} is the vector of output references known in advance and $\mathbb{U}_a$, $\mathbb{D}_a$, $\mathbb{Y}_a$ are polyhedral sets of constraints inputs, inputs' increments, and outputs respectively.

5.1.4 Previewing MPC$_{\text{LPV}}$ with Future Parameter Estimation for Asset Control

In the P-MPC$_{\text{LPV}}$ algorithm (5.4) for Asset Control (AC), the LPV system (5.2) is iteratively rebuilt by the Eq. (5.3) using the estimated or measured parameter vector ρ_k, considered constant in prediction. In order to improve the controller performances, the estimation of the future behaviour of the time-varying parameters has been introduced in the controller. Due to the linearized behaviour of the inner closed loop system provided by the P-MPC$_{\text{LPV}}$, a LTI model of the

parameters evolution ρ is identified by the analysis of input/output data

$$\begin{aligned} x_{\rho_{k+1}} &= A_\rho x_{\rho_k} + B_\rho u_{a_k} \\ \rho_k &= C_\rho x_{\rho_k} \end{aligned} \tag{5.5}$$

where $A_\rho \in \mathbb{R}^{n_{x_\rho} \times n_{x_\rho}}$, $B_\rho \in \mathbb{R}^{n_{x_\rho} \times n_{u_a}}$, $C_\rho \in \mathbb{R}^{n_\rho \times n_{x_\rho}}$ and $x_{\rho_k} \in \mathbb{R}^{n_{x_\rho}}$ is the parameter state vector. The goodness of the identified model (5.5) is indicated by a Normalized Root Mean Square Error (NRMSE) average value of 85%. This model can be used to estimate the future time-varying parameters value. Considering the minimization problem (5.4) and introducing (5.5) in the prediction equations (5.4b),(5.4c), the minimization problem becomes a nonlinear problem hard to solve in real-time. In order to maintain a QP formulation, the parameters model (5.5) is used to evaluate the future parameter evolution ρ_{k+i} using the optimization sequence computed at the previous time instances to approximate the future control effort $u_{a_{k+i}}$ by the estimated future plant inputs $u_{a_{k+i|k-1}} = u_{a_{k-1}} + \Delta u_{a_{k+i|k-1}}$. By the introduction of the approximated plant inputs, LPV model matrices (5.3) used to compute the plant prediction become

$$\begin{aligned} A_{k+j|k-1} &= \bar{A}_0 + \sum_{i=1}^{n_\rho} \bar{A}_i \rho_{i_{k+j|k-1}} \\ B_{k+j|k-1} &= \bar{B}_0 + \sum_{i=1}^{n_\rho} \bar{B}_i \rho_{i_{k+j|k-1}} \\ C_{k+j|k-1} &= \bar{C}_0 + \sum_{i=1}^{n_\rho} \bar{C}_i \rho_{i_{k+j|k-1}}. \end{aligned} \tag{5.6}$$

The parameter approximation is computed by

$$\rho_{k+j|k-1} = C_\rho A_\rho^j x_{\rho_k} + C_\rho A_\rho^{j-1} B_\rho \left(u_{a_{k-1}} + \sum_{l=1}^{j} \Delta u_{a_{k+h|k-1}} \right) \tag{5.7}$$

where, considering the parameter prediction horizon N_{f_a} such that $l \leq N_{f_a} \leq N_{p_a}$, the predicted increment input is $\Delta u_{a_{k+h|k-1}} = \Delta u_{a_{k+h-1|k-1}}$ for $l \geq N_{u_a}$ and the predicted LPV matrices are

$$\begin{aligned} &A_{k+j|k-1} = A_{k+j-1|k-1} \ , \ B_{k+j|k-1} = B_{k+j-1|k-1} \\ &C_{k+j|k-1} = C_{k+j-1|k-1} \text{ for } j \ \geq \ N_{f_a}. \end{aligned} \tag{5.8}$$

From Eq. (5.6), the prediction model considered in the optimization problem of Eq. (5.4) becomes

$$\begin{aligned} x_{a_{k+i+1|k}} &= A_{k+i|k-1} x_{a_{k+i|k}} + B_{k+i|k-1} u_{a_k+i|k} \\ y_{a_{k+i+1|k}} &= C_{k+i|k-1} x_{a_{k+i+1|k}} \end{aligned} \tag{5.9}$$

In the next, the previewing MPC_{LPV} with the prediction model provided by Eq. (5.9) is called $\text{P}_F\text{-MPC}_{\text{LPV}}$.

Table 5.3: Asset Controller Tuning

N_{p_a}	15
N_{u_a}	5
N_{r_a}	3
N_{f_a}	3
Q_a	$\begin{bmatrix} 500 & 500 & 1 & 500 \end{bmatrix}$
R_a	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
P_a	$\begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 2000 \end{bmatrix}$
Input constraints	$-0.1 \leq \delta_a \leq 0.1$ $-0.1 \leq \delta_r \leq 0.1$ $-0.1 \leq \delta_e \leq 0.1$ $0.1 \leq \delta_t \leq 1$
Output constraints	$-0.15 \leq \psi \leq 0.15$ $-0.15 \leq \phi \leq 0.15$ $-0.15 \leq \theta \leq 0.15$ $20 \leq V_a \leq 50$
T_a	400 ms

5.1.5 Switching Previewing Model Predictive Control for Altitude Control

For the altitude External Controller (EC), a Switching MPC (MPC_{S}) is proposed. The MPC_{S} policy, proposed in [141, 142], considers an optimization problem formulated as a mixed-integer quadratic program where the integer switching variables are used to select the prediction model. In the proposed approach, the considered switching controllers have been developed with respect

to a LTI prediction model. These controllers have been tuned with respect to the different fight control scenarios. The prediction model related to the altitude controller (EC) has been identified as a LTI system (NRMSE fit 88%)

$$\begin{aligned} x_{e_{k+1}} &= A_e x_{e_k} + B_e u_{e_k} \\ y_{e_k} &= C_e x_{e_k} \end{aligned} \tag{5.10}$$

with $r_{\theta_k} = u_{e_k} \in \mathbb{R}$, $x_{e_k} \in \mathbb{R}^{n_{x_e}}$. The considered minimization problem (P-MPC$_{\text{S}}$) is

$$\min_{\Delta u_a} \sum_{i=1}^{N_{p_e}-1} \|Q_{e_s}(y_{e_{k+i|k}} - r_{H_k+i})\|_2^2 + \sum_{l=0}^{N_{u_e}-1} \|R_{e_s} \Delta u_{e_{k+l|k}}\|_2^2 + \|P_{e_s}(y_{e_{k+N_{pe}|k}} - r_{H_{k+N_{pe}}})\|_2^2 \tag{5.11}$$

$$\text{s.t.} \quad x_{e_{k+i+1|k}} = A_e x_{e_{k+i|k}} + B_e u_{e_{k+i|k}} \tag{5.12}$$

$$y_{e_{k+i+1|k}} = C_e x_{e_{k+i+1|k}} \tag{5.13}$$

$$x_{e_{k|k}} = x_{e_k} \tag{5.14}$$

$$\Delta u_{e_{k+N_u+j|k}} = 0 \tag{5.15}$$

$$r_{H_{k+N_{r_a}+v}} = r_{H_{k+N_{r_e}+v-1}} \tag{5.16}$$

$$u_{e_{k+l|k}} \in \mathbb{U}_e \tag{5.17}$$

$$\Delta u_{e_{k+l|k}} \in \mathbb{D}_e \tag{5.18}$$

$$y_{e_{k+l|k}} \in \mathbb{Y}_e \tag{5.19}$$

$$i \in \{0, \ldots, N_{p_e} - 1\} \tag{5.20}$$

$$l \in \{0, \ldots, N_{u_e} - 1\} \tag{5.21}$$

$$j \in \{0, \ldots, N_{p_e} - N_{u_v} - 1\} \tag{5.22}$$

$$v \in \{0, \ldots, N_{p_e} - N_{r_e} - 1\} \tag{5.23}$$

$$s \in \{1, \ldots, N_s\} \tag{5.24}$$

where N_{p_e} is the prediction horizon, N_{u_e} is the control horizon, N_{r_e} is the previewing horizon, Q_{e_s}, R_{e_s}, P_{e_s} are weight matrices of appropriate dimension, s is the switching variable provided externally to set the switch time instance between controllers, N_s is the number of provided controllers, $x_{e_{k+i|k}}$ denotes the prediction of the variable x_H at time $k+i$ based on the information available at time k, $\Delta u_{e_{k+i|k}}$ is the vector of the input increments, with $u_{e_{k-1|k}} = u_{e_{k-1}}$, r_{H_k+i} is the vector of output references known in advance and $\mathbb{U}_e$, $\mathbb{D}_e$, $\mathbb{Y}_e$ are polyhedral sets of constraints inputs, inputs' increments, and outputs respectively. By the formulation of Eq. (5.11), P-MPC$_{\text{S}}$ can be consider as a P-MPC$_{\text{LPV}}$ rebuilding the QP when the variable s changes between consecutive time instances.

Table 5.4: Altitude Controllers Tuning

Common Controllers Parameters	
N_{p_e}	45
N_{u_e}	15
N_{r_e}	3
N_{c_e}	3
Input constraints	$-0.15 \leq u_e \leq 0.15$
Input rate constraints	$-0.025 \leq \Delta u_e \leq 0.025$
Output constraints	$1000 \leq h \leq 7000$
T_e	4 s
Controller Parameters EC_1	
Q_{e_1}	10
R_{e_1}	1000
P_{e_1}	1000
Controller Parameters EC_2	
Q_{e_2}	10
R_{e_2}	1000
P_{e_2}	10
Controller Parameters EC_3	
Q_{e_3}	100
R_{e_3}	1000
P_{e_3}	10

5.1.6 Smooth Switching Previewing Model Predictive Control for the Altitude Control

Considering the previous P-MPC$_S$ formulation, an algorithm name P$_S$-MPC$_S$ is proposed in order to obtain a smooth behaviour during the switching step between different tuned controllers. By the anticipate action [133], it is possible to consider known in advance the prescribed switching time instance. From the switching variable s, it is possible to define a switching vector $\bar{s} \in \mathbb{R}^{N_{c_e}}$, where N_{c_e} is the switching preview horizon $1 \leq N_{c_e} \leq N_{p_e}$ such that

$$\bar{s} = \begin{bmatrix} s_{k|k} & \dots & s_{k+N_s|k} \end{bmatrix}' \tag{5.25}$$

where $s_{k|k} = s_k$ is the actual required controller index and $s_{k+i|k}$ is the required controller known in advance at the switching time instance $k+i$ with $i = 1, \dots, N_{c_e}$. The cost function of the minimization problem (5.11) can be

redefined as

$$\min_{\Delta u_a} \sum_{i=1}^{N_{p_e}-1} \|Q_i(y_{e_{k+i|k}} - r_{H_k+i})\|_2^2 + \sum_{l=0}^{N_{u_e}-1} \|R_l \Delta u_{e_{k+l|k}}\|_2^2 + \|P_{e_{s_{k+N_{c_e}|k}}}(y_{e_{k+N_{p_e}|k}} - r_{H_{k+N_{p_e}}})\|_2^2 \tag{5.26}$$

with $Q_i = Q_{e_{s_{k+i|k}}}$ and $R_h = R_{e_{s_{k+h|k}}}$. Due to the previewing action, the weights matrices are introduced iteratively in the control law, gradually switching into the successive controller, and optimizing the control behaviour before the controller complete switch.

5.1.7 Simulation results: Asset Controllers comparison

In this section the comparison between the P$_F$-MPC$_{\text{LPV}}$ and the P-MPC$_{\text{LPV}}$ of Sec. 5.1.3 and 5.1.4, respectively, is reported. The controller parameters are given in Tab. 5.3. In the simulation the stabilization performance have been presented, showing the smooth behaviour of the controlled output due to the introduction of the time-varying parameters future behaviour estimation. Results related to several reference step variations are given in Tab. 5.5, reporting the average percentage ISE improvement

$$\Delta\text{ISE} = \frac{\text{ISE}_{\text{P-MPC}_{\text{LPV}}} - \text{ISE}_{\text{P}_F\text{-MPC}_{\text{LPV}}}}{\text{ISE}_{\text{P-MPC}_{\text{LPV}}}} \times 100 \tag{5.27}$$

provided by the P$_F$-MPC$_{\text{LPV}}$ with respect to the P-MPC$_{\text{LPV}}$. Fig.5.3 presents aileron δ_a, rudder δ_r, elevator δ_e and throttle δ_t control efforts and outputs (bank ϕ, heading ψ, pitch θ and airspeed V_a) controlled by the P-MPC$_{\text{LPV}}$ and the P$_F$-MPC$_{\text{LPV}}$ respectively. In Fig. 5.4 are reported the related parameters variation (Angle of Attack α, sideslip β, Mach speed M and atmospheric pressure p). The better performance improvement is related to the latitudinal subsystem controlled output (75% and 35% for ϕ and ψ respectively) with respect to the longitudinal subsystem output (8% and 3% for θ and V_a respectively). Infact, the introduced parameter prediction provides a considerable improvement only when the considered parameter has a dynamics predictable over the considered control horizon. This is the case of the parameter α that is featured by a faster dynamics during manoeuvring task, in the presence of a considerable input variation. Whereas a parameter featured by a slower dynamics can be assumed constant in prediction. As in the case of M and ρ, that do not present a considerable variation in the considered scenario.

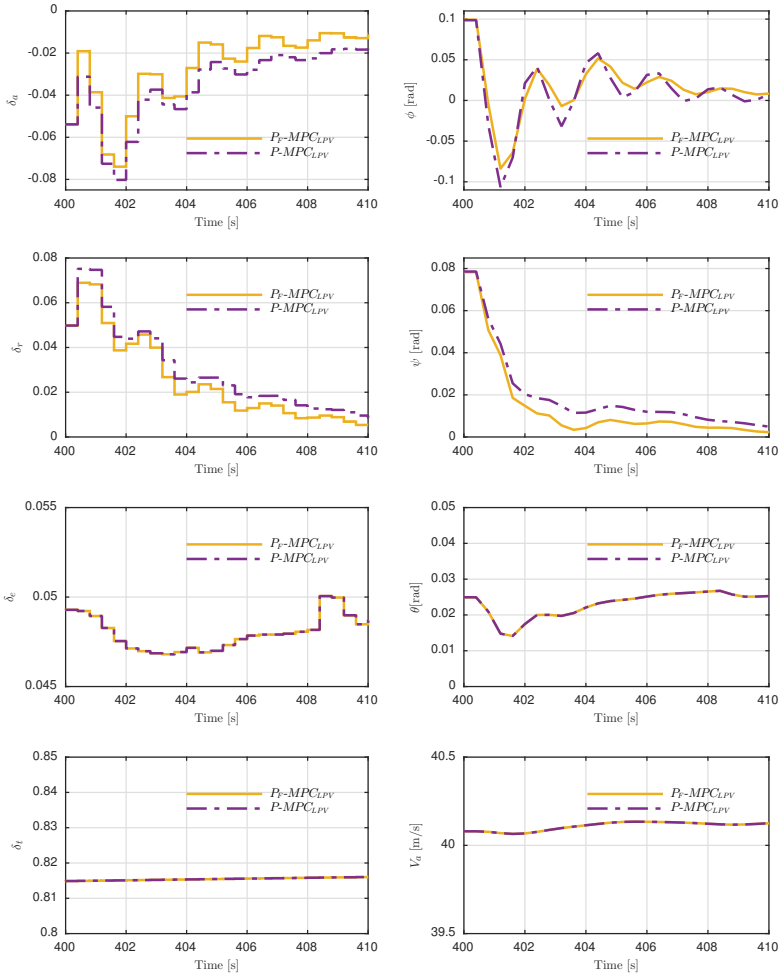

Figure 5.3: Comparison between P-MPC$_{\text{LPV}}$ and P$_F$-MPC$_{\text{LPV}}$ techniques, stabilization task scenario.

5.1.8 Simulation results: Altitude Controllers comparison

The comparison results between the P-MPC$_{\text{S}}$ and the P$_S$-MPC$_{\text{S}}$ of Sec. 5.1.5 and 5.1.6 have been reported. The control scenario consists of a change in the altitude reference, driving the Aerosonde from an initial altitude to a different one. In the proposed test the switch between the three controllers, i.e.

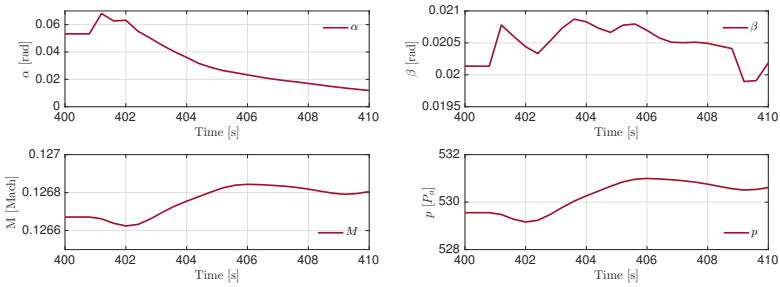

Figure 5.4: Scheduling Time-Varying parameters.

Table 5.5: Comparison of P-MPC$_{\text{LPV}}$ and P$_F$-MPC$_{\text{LPV}}$: Average Integral Square Error Results

	ϕ	ψ	θ	V_a
ΔISE	**45.69%**	**55.79%**	**13.13%**	**1.74%**

EC_1, EC_2 and EC_3, is tested considering the controller parameters reported in Tab. 5.4. Each controller is tuned in order to provide better performances with respect to the corresponding control scenario: considering two time instances T_1=300 s and T_2=400 s, before the T_1 instance the controller EC_1 is used, during the interval between T_1 and T_2 the controller EC_2 is selected and finally after T_2 the control law switches to EC_3. Fig. 5.5 and 5.6 presents the simulation results for different altitude trajectory showing the altitude control effort u_e, the reference r$_H$ and the controlled output. Average percentage ISE improvement, computed as in Eq. (5.27), has been used to compare the performances improvement introduce by the P$_S$-MPC$_{\text{S}}$ with respect to the P-MPC$_{\text{S}}$ in a set of standard switching control scenarios. The comparison results shown an average improvement of ΔISE of 24.21% due to the introduction of the P$_S$-MPC$_{\text{S}}$. Simulation results of Fig. 5.5 and 5.6 report the improvement related to the smooth switching technique. The control effort presents a smooth behaviour, reducing the controlled output overshoots during the switch step in both the considered trajectories.

5.2 Computational efficient LPV-MPC

The success of Model Predictive Control (MPC) is growing in many engineering fields, thanks to the ability of optimally regulating multivariable systems while

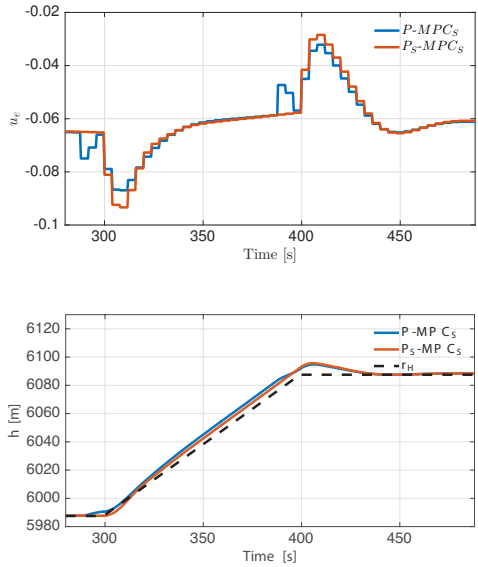

Figure 5.5: Comparison between P-MPC$_S$ and P$_S$-MPC$_S$ techniques controlling positive slope altitude trajectory.

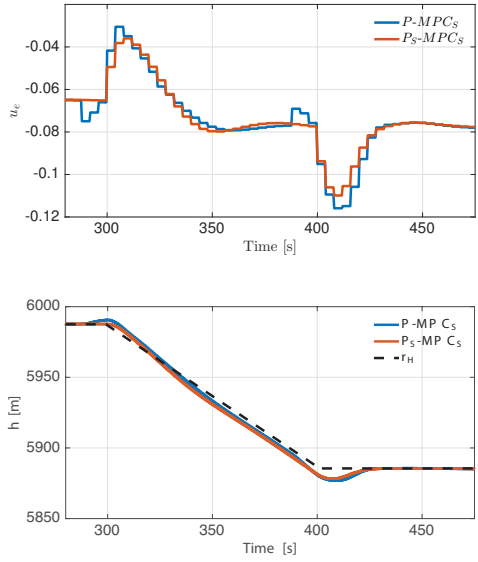

Figure 5.6: Comparison between P-MPC$_S$ and P$_S$-MPC$_S$ techniques controlling negative slope altitude trajectory.

imposing constraints on inputs and outputs. When the system to control has a nonlinear dynamics, the use of a Linear Parameter Varying (LPV) prediction

model has been proven to be an effective solution in many cases. However, the computational effort is a crucial issue for LPV-MPC, severely limiting its application especially in embedded control. Indeed, for small scale systems the time needed to compute the Quadratic Programming (QP) problem at each time step, can be substantially higher than the average time to solve it. In this section an algorithm that drastically reduces this computational complexity for a particular class of LPV systems is presented. It is shown that when the input matrix is right-invertible, the rebuild phase can be accelerated by means of a coordinate transformation which approximates the original problem. It is then introduced a variant of the algorithm, able to further reduce the rebuild time at the cost of a slightly increased sub-optimality. The two algorithm are tested on a benchmark set of random MPC problems, as well as on a case study regarding the control of an autonomous tram. Results on time complexity and optimality confirm the effectiveness of the method, especially in those applications where computational load is a key indicator for success. In Sec. 5.2.1 the problem statement is presented, introducing the reasons of the proposed research. Sections 5.2.2 and 5.2.2.1 present the algorithms formulation proposed and Sec. 5.2.3 introduced the benchmark of the proposed tests. The results of the simulation, considering the proposed case study are reported in Sec. 5.2.4.

5.2.1 Problem statement

Define $\theta_k \in \Theta$ a time-varying vector of exogenous parameters, measured at each time step k, with $\Theta \subset \mathbb{R}^{n_\theta}$ a set of interest. An LPV model can be written as follows:

$$x_{k+1} = A(\theta_k)x_k + B(\theta_k)u_k \tag{5.28a}$$

$$y_k = Cx \tag{5.28b}$$

where $x \in \mathbb{R}^{n_x}$ is the states' vector, $u \in \mathbb{R}^{n_u}$ is the inputs' vector, $y \in \mathbb{R}^{n_y}$ is the outputs' vector, $A(\theta_k) \in \mathbb{R}^{n_x \times n_x}$, $B(\theta_k) \in \mathbb{R}^{n_x \times n_u}$ and $C \in \mathbb{R}^{n_y \times n_x}$ are the state-space matrices. MPC_{LPV} makes use of (5.28) as a prediction model, and with a quadric cost function subject to affine constraints, the optimization

problem to solve can be formulated as:

$$\min_{\Delta u} \quad \sum_{i=1}^{N_p} \|Q_y(y_{k+i|k} - r_k)\|_2^2 + \sum_{h=0}^{N_u-1} \|Q_{\Delta u}\Delta u_{k+h|k}\|_2^2 \tag{5.29a}$$

$$\text{s.t.} \quad x_{k+i+1|k} = A(\theta_k)x_{k+i|k} + B(\theta_k)u_{k+i|k} \tag{5.29b}$$

$$y_{k+i+1|k} = Cx_{k+i+1|k} \tag{5.29c}$$

$$x_{k|k} = x_k \tag{5.29d}$$

$$\Delta u_{k+N_u+j|k} = 0 \tag{5.29e}$$

$$\Delta u_{k+h|k} \in \mathbb{D}_k \tag{5.29f}$$

$$y_{k+h|k} \in \mathbb{Y} \tag{5.29g}$$

$$i \in \{0, \dots, N_p - 1\} \tag{5.29h}$$

$$h \in \{0, \dots, N_u - 1\} \tag{5.29i}$$

$$j \in \{0, \dots, N_p - N_u - 1\}, \tag{5.29j}$$

where N_p is the prediction horizon, N_u is the control horizon, Q_y, and $Q_{\Delta u}$ are square weight matrices, with $Q_{\Delta u}$ invertible, $x_{k+i|k}$ denotes the prediction of the variable x at time $k+i$ based on the information available up to time k, $\Delta u_{k+i|k}$ is the vector of the input increments, with $u_{k-1|k} = u_{k-1}$, r_k is the vector of outputs references, and $\mathbb{D}_k$, $\mathbb{Y}$ are polyhedral sets of constraints inputs' increments, and outputs, respectively. Given the inputs constraints $E_u u \leq e_u$, and the inputs' increments constraints $E_{\Delta u}\Delta u \leq e_{\Delta u}$, then $\mathbb{D}_k = \{\Delta u \in \mathbb{R}^{n_u} \mid D\Delta u \leq d_k\}$ where $D = \begin{bmatrix} E_{\Delta u} & E_u \end{bmatrix}'$ and $d = \begin{bmatrix} e_{\Delta u} & e_u - E_u u_{k-1} \end{bmatrix}'$. The solution to (5.29) can be found by optimizing the equivalent time-varying QP problem below, often addressed as *condensed formulation*:

$$\begin{aligned} \min_{z} \quad & \frac{1}{2} z' H_k z + \rho_k' F_k' z \\ \text{s.t.} \quad & \begin{bmatrix} G^{\Delta u} \\ G_k^y \end{bmatrix} z \leq \begin{bmatrix} W^{\Delta u} \\ W_k^y \end{bmatrix} \rho_k + \begin{bmatrix} w^{\Delta u} \\ w_k^y \end{bmatrix} \end{aligned} \tag{5.30}$$

where $z \in \mathbb{R}^{n_z}$ is the vector of optimization variables, $\rho_k \in \mathbb{P}$ is the vector of MPC parameters of dimension $n_\rho = n_y + n_x + n_u$, with $\mathbb{P} \subset \mathbb{R}^{n_\rho}$ a bounded set of interest and $\rho_k = \begin{bmatrix} r_k & u_{k-1} & x_{k|k} \end{bmatrix}'$, $H_k \in \mathbb{R}^{n_z \times n_z}$ is a symmetric and positive definite matrix, $F_k \in \mathbb{R}^{n_z \times n_\rho}$ is the linear cost term, $G_k^{\Delta u} \in \mathbb{R}^{m_u \times n_z}$, $W_k^{\Delta u} \in \mathbb{R}^{m_u \times n_\rho}$ and $w_k^{\Delta u} \in \mathbb{R}^{m_u}$ are the terms defining inputs' increments constraints, $G_k^y \in \mathbb{R}^{m_y \times n_z}$, $W_k^y \in \mathbb{R}^{m_y \times n_\rho}$ and $w_k^y \in \mathbb{R}^{m_y}$ are the terms defining outputs constraints. The optimization variables are defined as $z = \begin{bmatrix} \Delta u_k, \dots, \Delta u_{k+N_u} \end{bmatrix}$. It is worth noticing that, as a standard assumption, the vector of parameters θ_k is supposed to be constant in prediction [44,45]. The recent literature proposed

several algorithms to solve efficiently the QP problem (5.30), among which the gradient projection [143], the interior-point [144] and active-set methods [145, 146]. However, solving problem (5.30) is not the only computationally costly operation for MPC_{LPV}. Indeed, unlike MPC_{LTI}, the matrices H_k, F_k, G_k, W_k and w_k are time-varying, because of the dependence on the model (5.28), and therefore they must be recomputed at each time step. Unfortunately, the time required by this rebuild can be high enough to discourage the use of MPC_{LPV}. As an example, Fig. 5.7 provides an idea about this complexity, comparing the times required to build the problem and to solve it, on a large set of random QP problems with dimensions commonly found in embedded applications. Clearly, the rebuild time depends only on the dimensions, whereas the solution time depends on the iterations taken by the solver and it is represented by a box-plot resulting from 1000 random tests for each point. The computational framework is a PC with Intel® Core i5-3210M CPU @ 2.50GHz, running the dual simplex solver from Mosek Aps©. The results show that the time spent on building the QP is substantially larger than the average time to solve it, and can be even larger than the worst-case.

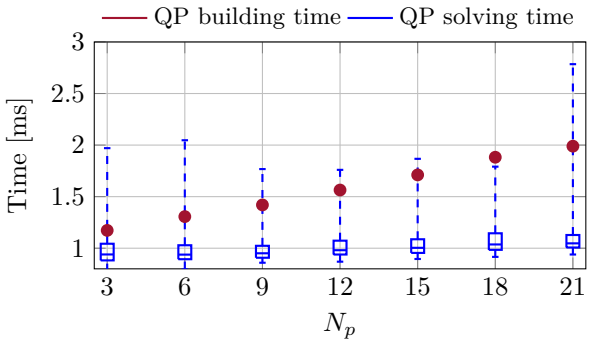

Figure 5.7: Computational time required to compute the optimal inputs with a standard MPC_{LPV} formulation with $n_y = n_x = n_u = 2$, $N_u = N_p/3$. The time required to compute the QP and to solve it are shown.

5.2.2 Rebuild-free LPV-MPC

From the considerations of previous section, it is easy to conclude that by reducing the time spent building the QP problem from scratch, the efficiency of MPC_{LPV} can be considerably improved. This section details the RF-MPC_{LPV} algorithm and shows the improvements when compared with standard MPC_{LPV}. As a second contribution, the RF_2-MPC_{LPV} variant is also presented, where, by scarifying solution accuracy, the speed can be further enhanced.

5.2.2.0.1 Coordinate transformation of system's inputs To derive the RF-MPC$_{\text{LPV}}$ framework, here a coordinate transformation of the systems' inputs has been introduced, which allows to design the MPC controller for a LTI system. This is close in spirit with Feedback Linearization (FL) technique, [147–150]. However, the drawbacks of FL are avoided in this case, as *i*) the cost function remains convex, *ii*) the constraints on the transformed variables can be computed by affine operations on the original ones. The following Theorem introduces the coordinate transformation, and derives the relations to correctly impose the constraints on the transformed variables.

Theorem 5.2.1. *Given an MPC formulation as in (4.37), with $n_u \geq n_x$, let $A(\theta_k) = A(\theta_0) + \bar{A}(\theta_k)$ and $B(\theta_k) = B(\theta_0) + \bar{B}(\theta_k)$, two time-varying matrices computed at time k, for some $\theta_0 \in \Theta$. Consider the LTI system defined by*

$$\bar{x}_{k+1} = A(\theta_0)\bar{x}_k + B(\theta_0)\bar{u}_k \tag{5.31a}$$

$$\bar{y}_k = C\bar{x}_k. \tag{5.31b}$$

and let $\mathcal{F}(\bar{u}_k, x_k) : \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$ be a coordinate transformation of the inputs defined as

$$\mathcal{F}(\bar{u}_k, x_k) := u_k = B(\theta_k)^+(B(\theta_0)\bar{u}_k - \bar{A}(\theta_k)x_k) \tag{5.32}$$

with $B(\theta_k)^+$ the right pseudo-inverse of $B(\theta_k)$. Then, to impose constraints on the original LPV system (5.28), the LTI system (5.31) must be constrained such as:

i) given $\Delta u \in \mathbb{D}_k$, then $\Delta\bar{u}_{k+h} \in \bar{\mathbb{D}}_{k+h}$ where $\bar{\mathbb{D}}_{k+h}$ is an affine transformation of $\mathbb{D}_k$, depending on x_{k+h};

ii) given $y_{k+i} \in \mathbb{Y}$, then $\bar{y}_{k+i} \in \mathbb{Y}$, $\forall\, i = 0, \ldots, N_p - 1$.

Proof 1. *i) Consider $\mathbb{D}_k = \{\Delta u \in \mathbb{R}^{n_u} \mid D\Delta u \leq d_k\}$, then applying the coordinates' transformation (5.32), the following polyhedral space is derived*

$$\bar{\mathbb{D}}_{k+h} = \{\Delta\bar{u} \in \mathbb{R}^{n_u} | \bar{D}_{k+h}\Delta\bar{u} \leq \bar{d}_{k+h},\, \forall h = 0 \ldots N_u\} \tag{5.33}$$

where

$$\begin{aligned} \bar{D}_{k+h} &= DB(\theta_k)^+B(\theta_0) \\ \bar{d}_{k+h} &= DB(\theta_k)^+\bar{A}(\theta_k)x_{k+h|k} + d_k. \end{aligned} \tag{5.34}$$

ii) Given $u_{k+i|k} = \mathcal{F}(\bar{u}_{k+i|k}, x_{k+i|k})$, $\forall i = 0, \ldots, N_p - 1$, with $u_{k+i|k} = u_{k+N_u-1|k}$, $\forall i \geq N_u$. Let the output prediction equation be

$$y_{k+i+1|k} = C(A(\theta_k)x_{k+i|k} + B(\theta_k)u_{k+i|k}), \tag{5.35}$$

then, applying the coordinate transformation, the following equation is obtained:

$$y_{k+i+1|k} = C(A(\theta_0)x_{k+i|k} + B(\theta_0)\bar{u}_{k+i|k}) \tag{5.36}$$

and therefore $y_{k+i+1|k} = \bar{y}_{k+i+1|k}$ *holds, from which clearly follows that* $\bar{y}_{k+i} \in \mathbb{Y}$. *As a by-product, equation (5.36) demonstrates also that* $x_{k+i|k} \equiv \bar{x}_{k+i|k}$ *which will be useful in the next sections.*

Theorem 5.2.1 suggests how to design a MPC controller for the LTI system (5.31), that, when used in feedback with the coordinates' transformation (5.32), regulates the original LPV system by imposing the desired inputs/outputs constraints. In the specific, given the proved equivalence of the outputs, the outputs' constraints can be imposed directly on the LTI system, without the need of time-dependent rebuild in the condensed form. On the other hand, the constraints on the inputs can be derived by an affine transformation of the original set. However, this transformation would depend on the future evolution of the states, which would turn into a non-holonomic constraints' formulation, difficult to be solved in real-time. The next subsection proposes the complete RF-MPC$_{\text{LPV}}$ algorithm, with a solution to solve efficiently the aforementioned problem.

The convex optimization problem to be solved by the MPC controller, designed to regulate the LTI system (5.31), can be cast into the following QP problem, where the Hessian H, the matrix F, and the outputs' matrix inequality are time-invariant:

$$\begin{aligned} &\min_{\bar{z}} \ \frac{1}{2}\bar{z}'H\bar{z} + \rho_k'F'\bar{z} \\ &\text{s.t.} \ \begin{bmatrix} G^{\Delta u}\bar{G}_k \\ G_0^y \end{bmatrix} \bar{z} \leq \begin{bmatrix} T_k \\ W_0^y \end{bmatrix} \rho_k + \begin{bmatrix} \bar{T}_k \\ 0^{(m_y \times n_u)} \end{bmatrix} \bar{\rho}_k + \begin{bmatrix} w^{\Delta u} \\ w_0^y \end{bmatrix} \end{aligned} \tag{5.37}$$

with $\bar{z} = \left[\Delta\bar{u}_k, \ldots, \Delta\bar{u}_{k+N_u}\right]$, $F = F(\theta_0)$, $H = H(\theta_0)$, $G_0^y = G^y(\theta_0)$, $W_0^y = W^y(\theta_0)$, $w_y^0 = w^y(\theta_0)$, and T_k, $\bar{T}_k$, $\bar{G}_k$ and $\bar{\rho}_k$ are the only time-varying components whose derivation is detailed below. Consider $G_{h,j}^{\Delta u}$ the row of $G^{\Delta u}$ relative the the j-th input increment constraint, at the h-th prediction step, with $j = 1, \ldots, m_u$ and $h = 0, \ldots, N_u - 1$. Assume $G^{\Delta u}$ to be ordered by grouping all the constraints at the same prediction step, with increasing j, and let $w^{\Delta u}$ follow the same order scheme, with $w_{h,j}^{\Delta u}$ the single element. By applying the coordinates' transformation (5.32), for $h = 0$ the input increments can be written as:

$$\Delta u_{k|k} = B(\theta_k)^+(B(\theta_0)\bar{u}_{k|k} - \bar{A}(\theta_k)x_{k|k}) - u_{k-1} \tag{5.38}$$

and the j-th input constraint on the new coordinates can be computed as

$$\bar{T}^{\bar{u}}_{0,j}\Delta\bar{u}_{k|k} \leq w_{0,j} - \bar{T}^{\bar{u}}_{0,j}\bar{u}_{k-1} + G^{\Delta u}_{0,j}u_{k-1} + \bar{T}^{x}_{0,j}\bar{x}_{k|k} \tag{5.39}$$

with

$$\bar{T}^{\bar{u}}_{0,j} = G^{\Delta u}_{0,j}B(\theta_k)^{+}B(\theta_0) = G^{\Delta u}_{0,j}\bar{G}_{0,j_k} \tag{5.40a}$$

$$\bar{T}^{x}_{0,j} = G^{\Delta u}_{0,j}B(\theta_k)^{+}\bar{A}(\theta_k) \tag{5.40b}$$

Equation (5.38) allows to recompute the inputs' increments (and inputs) constraints in the new coordinates, to *exactly* impose those limits on the original LPV system. As a result, the inputs effectively applied to the system will be always feasible. However, the constraints at the future prediction steps are non-holonomic, due to what proved by Theorem 5.2.1. Therefore, in order to maintain the convexity of the problem, RF-MPC$_{\text{LPV}}$ approximates the prediction of the states with that computed at the previous time-step $k-1$. With this assumption, the general constraint j, at prediction step $v = 1, \ldots, N_u$, can be approximated by:

$$\begin{aligned}\bar{T}^{\bar{u}}_{v,j}\Delta\bar{u}_{k+v|k} \leq w_{v,j} - \bar{T}^{\bar{u}}_{v,j}\bar{u}_{k-1} + G^{\Delta u}_{0,j}u_{k-1} + \\ + \bar{T}^{x}_{v,j}\bar{x}_{k+v|k-1}\end{aligned} \tag{5.41}$$

where $x_{k+v|k-1}$ denotes the states' prediction obtained at the previous time-step, which can be derived by considering:

$$\bar{x}_{k+v|k-1} = A(\theta_0)\bar{x}_{k+v-1|k-1} + B(\theta_0)\bar{u}_{k+v-1|k-1} \tag{5.42}$$

with $\bar{x}_{k|k-1} = \bar{x}_{k|k}$, and $\bar{u}_{k+N_u-1|k-1} = \bar{u}_{k+N_u-2|k-1}$ due to the unavailability of $\Delta\bar{u}_{k+N_u-1|k-1}$. Consequently, the prediction of the LPV inputs are computed as:

$$\begin{aligned}u_{k+v-1|k-1} = & B(\theta_k)^{+}B(\theta_0)\bar{u}_{k+v-1|k-1} + \\ & - B(\theta_k)^{+}\bar{A}(\theta_k)\bar{x}_{k+v-1|k-1}.\end{aligned} \tag{5.43}$$

The equations (5.39) and (5.41) are the ground for the derivation of the time-varying components of the constraints' inequality in (5.37). In the specific, $\bar{\rho}_k$

is found to be $\bar{\rho}_k = \begin{bmatrix} \bar{\rho}_k^u & \bar{\rho}_k^{\bar{u}} & \bar{\rho}_k^x \end{bmatrix}'$ with

$$\bar{\rho}_k^u = \begin{bmatrix} \bar{u}_{k-1|k-1} & \dots & \bar{u}_{k+N_u-1|k-1} & 0^{(n_u \times 1)} \end{bmatrix} \tag{5.44a}$$

$$\bar{\rho}_k^{\bar{u}} = \begin{bmatrix} u_{k|k-1} & \dots & u_{k+N_u-1|k-1} & 0^{(n_x \times 1)} \end{bmatrix} \tag{5.44b}$$

$$\bar{\rho}_k^x = \begin{bmatrix} x_{k+1|k-1} & \dots & x_{k+N_p-1|k-1} \end{bmatrix}. \tag{5.44c}$$

As a last step, the time-varying components $\bar{G}_k$, T_k and $\bar{T}_k$ of the QP problem (5.37) are computed at each iteration as:

$$\bar{G}_k = I \begin{bmatrix} \bar{\mathrm{G}}_{0_k} & \dots & \bar{\mathrm{G}}_{h_k} & \dots & \bar{\mathrm{G}}_{N_u-1_k} \end{bmatrix} \tag{5.45a}$$

$$T_k = \begin{bmatrix} 0^{(m_u \times n_y)} & T_0^u & T_{0_k}^x \end{bmatrix} \tag{5.45b}$$

$$\bar{T}_k = \begin{bmatrix} \bar{T}_k^{\bar{u}} & \bar{T}^u & \bar{T}_k^x \end{bmatrix} \tag{5.45c}$$

$$\bar{T}_k^{\bar{u}} = I \begin{bmatrix} \bar{T}_{0_k}^{\bar{u}} & \dots & \bar{T}_{h_k}^{\bar{u}} & \dots & \bar{T}_{N_u-1_k}^{\bar{u}} \end{bmatrix} \tag{5.45d}$$

$$T^u = I \begin{bmatrix} 0^{(m_u \times n_u)} & T_1^u & \dots & T_h^u & \dots & T_{N_u-1}^u \end{bmatrix} \tag{5.45e}$$

$$T_k^x = I \begin{bmatrix} 0^{(m_u \times n_x)} & T_{1_k}^x & \dots & T_{h_k}^x & \dots & T_{N_u-1_k}^x \end{bmatrix} \tag{5.45f}$$

$$\bar{\mathrm{G}}_{h_k} = \begin{bmatrix} \bar{\mathrm{G}}_{h,1_k} \\ \vdots \\ \bar{\mathrm{G}}_{h,j_k} \\ \vdots \\ \bar{\mathrm{G}}_{h,m_{u_k}} \end{bmatrix}, \bar{T}_{h_k}^{\bar{u}} = \begin{bmatrix} -\mathrm{G}_{h,0}^{\Delta u} \bar{G}_{0,0_k} \\ \vdots \\ -\mathrm{G}_{h,j}^{\Delta u} \bar{G}_{0,j_k} \\ \vdots \\ -\mathrm{G}_{h,m_u}^{\Delta u} \bar{G}_{0,m_{u_k}} \end{bmatrix} \tag{5.45g}$$

$$T_h^u = \begin{bmatrix} \mathrm{G}_{h,0}^{\Delta u} \\ \vdots \\ \mathrm{G}_{h,j}^{\Delta u} \\ \vdots \\ \mathrm{G}_{h,m_u}^{\Delta u} \end{bmatrix}, T_{h_k}^x = \begin{bmatrix} \bar{T}_{h,0}^x \\ \vdots \\ \bar{T}_{h,j}^x \\ \vdots \\ \bar{T}_{h,m_u}^x \end{bmatrix} \tag{5.45h}$$

Given the above equations, Algorithm 2 details the steps of one iteration of RF-MPC$_{\text{LPV}}$.

5.2.2.1 Approximation with constraints constant in prediction

The RF-MPC$_{\text{LPV}}$ algorithm saves the costly rebuild of the hessian H_k, the matrix F_k and the matrix inequality of the output constraints. However, it still needs to rebuild at each time step the matrix inequality regarding the constraints on the inputs, and inputs' increments. This is unavoidable to have an

Algorithm 2 RF-MPC$_{\text{LPV}}$ Algorithm

Input: Parameters arrays ρ_k, θ_k, system state x_k, previous step inputs u_{k-1},$\bar{u}_{k-1}$, previous step input increments $\Delta u_{k+h|k-1}$, $h = 0, \ldots, N_u - 1$

$x_{k|k-1} \leftarrow x_k$;
$\bar{u}_{k-1|k-1} \leftarrow \bar{u}_{k-1}$;
$u_{k-1|k-1} \leftarrow u_{k-1}$;
$\bar{\rho}_k \leftarrow \emptyset$;
for $i = 1, \ldots, N_p$ **do**
    if $i \leq N_u - 1$ **then**
        $\bar{u}_{k+h|k-1} \leftarrow \bar{u}_{k+h-1|k-1} + \Delta\bar{u}_{k+h+1|k-1}$;
    else
        $\bar{u}_{k+h|k-1} \leftarrow \bar{u}_{k+h-1|k-1}$;
    end if
    compute $x_{k+h|k-1}$; ▷ See Eq. (5.42)
    $u_{k+h|k-1} \leftarrow \mathcal{F}\left(\bar{u}_{k+h|k-1}, x_{k+h|k-1}\right)$;
    $\bar{\rho}_k^u \leftarrow u_{k+h|k-1}$;
    $\bar{\rho}_k^x \leftarrow x_{k+h|k-1}$;
    $\bar{\rho}_k^{\bar{u}} \leftarrow \bar{u}_{k+h|k-1}$;
end for
$\bar{\rho}_k \leftarrow [\bar{\rho}_k^u, \bar{\rho}_k^x, \bar{\rho}_k^{\bar{u}}]'$;
build $\bar{G}_k$; ▷ See Eq. (5.45a)
build T_k; ▷ See Eq. (5.45b)
build $\bar{T}_k$; ▷ See Eq. (5.45c)
compute $\bar{z}^*$ by solving the QP problem (5.37);
$\bar{u}_k \leftarrow \bar{u}_{k-1} + \Delta\bar{u}_{k|k-1}$;
$u_k \leftarrow \mathcal{F}\left(\bar{u}_k, x_k\right)$;

Output: The inputs u_k to apply to the system.

accurate affine approximation of the original non-holonomic constraints, which translates into an optimizer very close to what obtained with the standard MPC$_{\text{LPV}}$. However, one can have the necessity to further improve the speed of the algorithm, at the cost of scarifying the solution accuracy. To this end, a simple but effective solution is to build the constraints at the current iteration step, that is equation (5.39), and impose it for the whole control horizon. This reflects a similar idea used in FL to approximate the non-linear constraints at future prediction time-steps, [148]. Therefore, in order to implement this variation of the algorithm, which is referred to as RF$_2$-MPC$_{\text{LPV}}$, matrices T_k,

T^u and $\bar{T}_k$ needs to be redefined, such as $\bar{\rho}_k = \bar{u}_{k-1}$ and

$$T_k = \begin{bmatrix} 0^{(N_u \cdot m_u \times n_y)} & T^u & T_k^x \end{bmatrix} \tag{5.46a}$$

$$T^u = \begin{bmatrix} T_0^u & \dots & T_h^u & \dots & T_{N_u-1}^u \end{bmatrix}' \tag{5.46b}$$

$$T_k^x = \begin{bmatrix} T_{0_k}^x & \dots & T_{h_k}^x & \dots & T_{N_u-1_k}^x \end{bmatrix}' \tag{5.46c}$$

$$\bar{T}_k = \begin{bmatrix} \bar{T}_{0_k}^{\bar{u}} & \dots & \bar{T}_{h_k}^{\bar{u}} & \dots & \bar{T}_{N_u-1_k}^{\bar{u}} \end{bmatrix}'. \tag{5.46d}$$

with T_h^u, $T_{h_k}^x$ and $\bar{T}_{h_k}^{\bar{u}}$ given by Eq. (5.45g) and (5.45h). Eq. (5.46) shows that the time-varying matrices required to update the QP problem (5.37) have a simpler structure for RF$_2$-MPC$_{\text{LPV}}$, with respect to RF-MPC$_{\text{LPV}}$. Moreover the states' approximation steps are not required. Both these aspects makes RF$_2$-MPC$_{\text{LPV}}$ a faster algorithm, at the price of an increased sub-optimality.

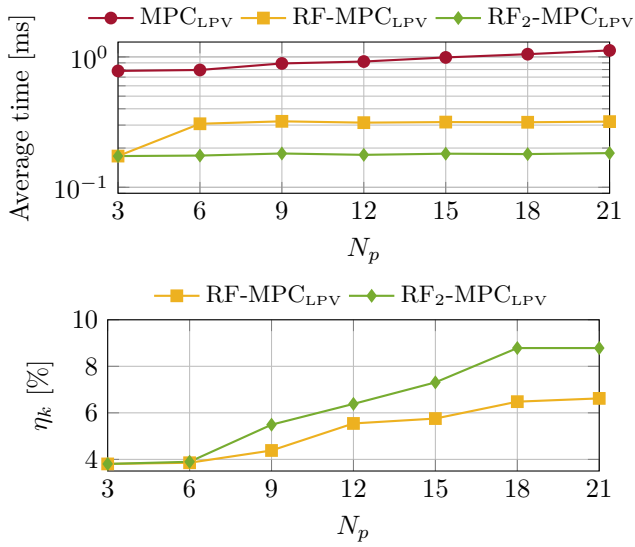

Figure 5.8: Comparison between MPC$_{\text{LPV}}$, RF-MPC$_{\text{LPV}}$ and RF$_2$-MPC$_{\text{LPV}}$ techniques when controlling systems with $n_y = n_x = n_u = 2$ and horizons $N_u = N_p/3$.

5.2.3 Benchmark on embedded problems

The proposed RF-MPC$_{\text{LPV}}$ and RF$_2$-MPC$_{\text{LPV}}$ have been tested on a large dataset of MPC problems, subject to box constraints on inputs, inputs' rate and outputs. The dimensions of such problems are typical of embedded applications, given the impact of the algorithm on real-time systems. The perfor-

mances have been compared with respect to the standard MPC_{LPV} approach, in terms of speed and solution accuracy. The computational environment is the one introduced in section 5.2.1. In order to evaluate the optimality, let the cost at instant k be

$$J_k = 0.5 z_k' H_k z_k + \rho_k F_k' z_k. \tag{5.47}$$

Given J_k^* the cost at z_k^*, which is the optimizer computed by standard MPC_{LPV}, the optimality gap η_k is introduced:

$$\eta_k = 100 \cdot |J_k^* - \hat{J}_k| \,/\, |J_k^*| \tag{5.48}$$

with $\hat{J}_k$ the cost computed at $\hat{z}^*$, defined as the optimizer obtained with either RF-MPC_{LPV} or RF_2-MPC_{LPV}. Figures 5.8 and 5.9 collect the results of the benchmark. Each single point is the average value of 1000 randomly generated MPC problems. It is worth noticing that the time reported refers to the whole control routine. Figure 5.8 shows how the algorithm performs with increasing prediction and control horizons. As expected, RF-MPC_{LPV} drastically reduces the computational time of MPC_{LPV}, from 7 to 8 times less, with a solution quality close to the optimum (less than 7% error in the worst-case). On the other hand, RF_2-MPC_{LPV} further accelerates the control routine, by allowing an higher sub-optimality. Figure 5.9 proposes the dependence of the results from the systems dimensions, keeping constant control and prediction horizon. Similar speed improvements and optimality gap can be observed in this case, too.

5.2.4 A Case Study

This section shows the noteworthy performance of the proposed algorithms on a real-world example of industrial relevance [151]. The control problem regards the low-level automatic train operation system of a catenary-free tram with regenerative breaking. In brief, the electrical machine and braking forces have to be optimally actuated to move the tram between two consecutive stations, guaranteeing punctuality, comfort and energy efficiency. The system dynamics can be represented by the following nonlinear, continuous-time, model:

$$\dot{s}(t) = v(t) \tag{5.49a}$$

$$\dot{v}(t) = \frac{f_e(t) + f_f(t) - 0.5\rho C_{DA} v(t)^2}{M} - h(\beta(t)) \tag{5.49b}$$

$$h(\beta(t)) = \mu g \cos(\beta(t)) + g \sin(\beta(t)) \tag{5.49c}$$

where s is the tram position to be controlled, v the tram speed, M the mass, f_e and f_f the electrical machine and braking forces, respectively, C_{DA} the

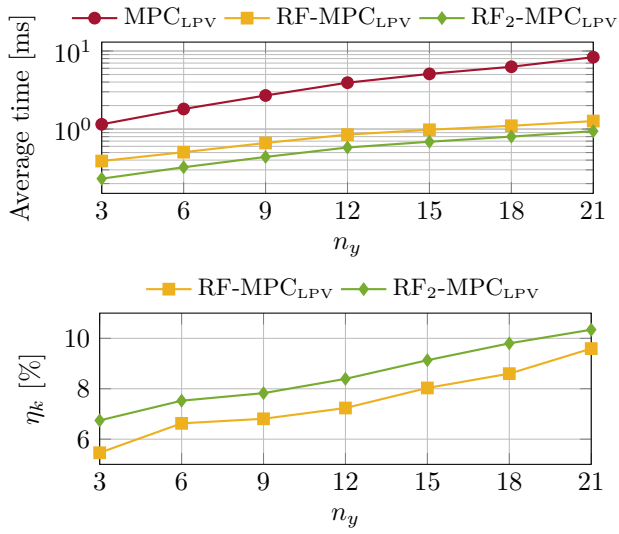

Figure 5.9: Comparison between MPC$_{\text{LPV}}$, RF-MPC$_{\text{LPV}}$ and RF$_2$-MPC$_{\text{LPV}}$ techniques consider plants with $n_y = n_x = n_u$ and horizons $N_p = 12$, $N_u = 4$.

coefficient of aerodynamic drag, ρ the density of air, $\beta(t)$ the grade angle, μ the coefficient of rolling distance, and g the gravity acceleration. Let us define $\alpha = 0.5\rho C_{DA}$, and let $a(t) = \dot{v}(t)$ be the acceleration of the tram, the vehicle dynamics must be subject to the following set of time-varying constraints on speed, acceleration, electrical machine and friction forces:

$$0 \leq v(t) \leq v^{\max}(s(t)) \tag{5.50a}$$
$$a^{\min} \leq a(t) \leq a^{\max} \tag{5.50b}$$
$$f_e^{\min}(v(t)) \leq f_e(t) \leq f_e^{\max}(v(t)) \tag{5.50c}$$
$$f_f^{\min}(t) \leq f_f(t) \leq 0 \tag{5.50d}$$
$$f_e(t) + f_f(t) \geq Ma^{\min} + \alpha v(t)^2 + Mh(\beta(t)) \tag{5.50e}$$

where, given the motor power P_m, the electrical force limits can be computed on-line as:

$$f_e^{\max}(v(t)) = \min\left(\frac{P_m^{\max}}{v(t)}, f_{e,a}^{\max}(t)\right) \tag{5.51a}$$
$$f_e^{\min}(v(t)) = \frac{P_m^{\min}}{v(t)} \tag{5.51b}$$
$$f_{e,a}^{\max}(t) = Ma^{\max} + \alpha v(t)^2 + Mh(\beta), \tag{5.51c}$$

an the time-varying limit $v_{max}(s(t))$ is a piece-wise constant function of the position. The meaning and values assumed by all the parameters in Eqs. (5.50) and (5.51) are collected in Table 5.6. It has been demonstrated as the non-linear dynamics 5.49 of the vehicle can be approximated by an LPV model, with sufficient accuracy for control purposes [151]. Let us define $x(t) = \begin{bmatrix} s(t) & v(t) \end{bmatrix}'$, $u(t) = \begin{bmatrix} f_e(t) & f_f(t) \end{bmatrix}'$ and $u_v(t) = \begin{bmatrix} \sin(\beta) & \cos(\beta) \end{bmatrix}'$, the following LPV system is used as the prediction model for MPC:

$$\dot{x}(t) = A^c(x_2(t))x(t) + B_u^c u(t) + B_v^c u_v(t), \tag{5.52}$$

with the continuous-time LPV matrices defined as:

$$A^c(x_t) = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{ax_2(t)}{M} \end{bmatrix}, \; B_u^c = \begin{bmatrix} 0 & 0 \\ \frac{1}{M} & \frac{1}{M} \end{bmatrix} \tag{5.53a}$$

$$B_v^c = \begin{bmatrix} 0 & 0 \\ -\mu g & -g \end{bmatrix}, \; C^c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.53b}$$

The discrete-time model is easily obtained, such that:

$$A^d(\theta_k) = e^{A^c(\theta(t))T_s} \tag{5.54a}$$

$$B_u^d(\theta_k) = \int_0^{T_s} e^{A^c(\theta(t))T_s} B_u^c d\tau \tag{5.54b}$$

$$B_v^d(\theta_k) = \int_0^{T_s} e^{A^c(\theta(t))T_s} B_v^c d\tau \tag{5.54c}$$

$$C^d \equiv C^c. \tag{5.54d}$$

Model (5.54) with constraints (5.50) are the bricks for the development of the MPC controller to be tested. Regarding the design of such controller, the reader is pointed to [151], which details the specifics of the MPC_{LPV} controller which is used in the following as the baseline to compare the novel algorithms. The main parameters of the MPC are collected in the second half of Table 5.6. It is worth mentioning that the tested controller makes use of the preview both on the reference position and on the measured disturbances. In order to derive the corresponding RF-MPC_{LPV} and RF_2-MPC_{LPV} controllers, the value $\theta_0 = 10$ m/s of the time-varying parameter $x_2(t)$ is selected, being the average value in the allowed range described by the speed constraints in the considered scenario. The comparison of the control performance between MPC_{LPV}, RF-$\text{MPC}_{\text{LPV}}$and RF_2-MPC_{LPV}, when driving the tram between to consecutive stations, is presented in Figures 5.10 and 5.11. They show the input forces computed by the three algorithms, the tracked output position and the output vehicle speed with related constraints, respectively. Clearly, the differences in

the overall control performance is not relevant, but the improvement in the average computational T_a time and maximum computational time T_M is substantial as confirmed by Table 5.7. With an average sub-optimality of η_a of 9%, the proposed techniques are able to reduce the average percentage computation ΔT_a time up to the 53%. It is worth mentioning that the error in the solution is computed on the whole, predicted, control sequence. However, only the first inputs are applied and RF-MPC$_{\text{LPV}}$ produces a maximum error on the actual inputs of 1.04% for u_1 and 0.38% for u_2. Whereas, for RF$_2$-MPC$_{\text{LPV}}$ this error amounts to 2.8% and 1.6%, for u_1 and u_2 respectively.

Table 5.6: Tram Specifications and MPC Tuning Parameters

Parameter	Symbol	Value	Units
Mass	M	80000	kg
Rolling resistance coeff.	μ	0.005	
Aerodynamic drag coeff.	C_{DA}	5.4	
Air density	ρ	1	
Max. acceleration	$a^{\max}$	1	m/s^2
Min. acceleration	$a^{\min}$	-1.2	m/s^2
Max. motor power	$P_m^{\max}$	850	kW
Min. motor power	$P_m^{\min}$	850	kW
Min. brake force	$f_f^{\min}$	10^5	N
Prediction horizon	N_p	20	
Control horizon	N_u	11	
output weights	Q_y	$\begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}$	
input weights	Q_u	$\begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}$	
input rates weights	$Q_{\Delta u}$	$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$	
Sampling time	T_s	0.2	s

Table 5.7: Tram position tracking with MPC- Performance Comparison

	MPC$_{\text{LPV}}$	RF-MPC$_{\text{LPV}}$	RF$_2$-MPC$_{\text{LPV}}$
T_a [s]	0.0123	0.00599	0.00574
T_M [s]	0.0404	0.01771	0.01481
ΔT_a [%]	$--$	51.4525	53.5225
η_a[%]	$--$	9.8280	9.8297

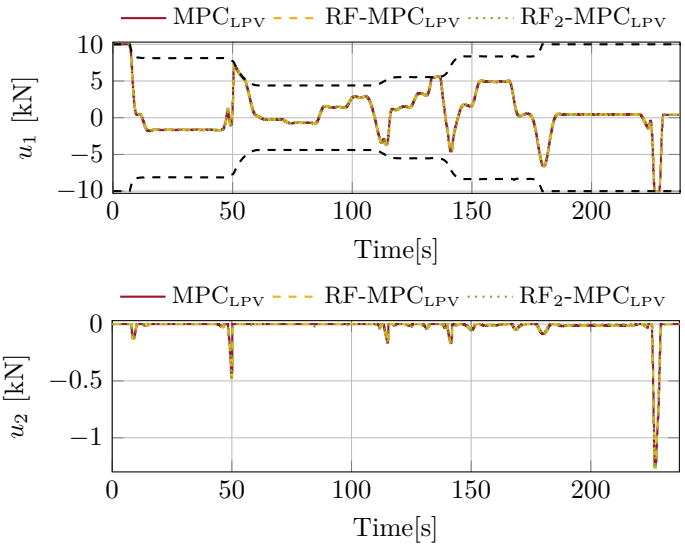

Figure 5.10: Comparison of input forces u_1 and u_2 when driving the tram with MPC$_{\text{LPV}}$, RF-MPC$_{\text{LPV}}$and RF$_2$-MPC$_{\text{LPV}}$ algorithms together with the input constraints.

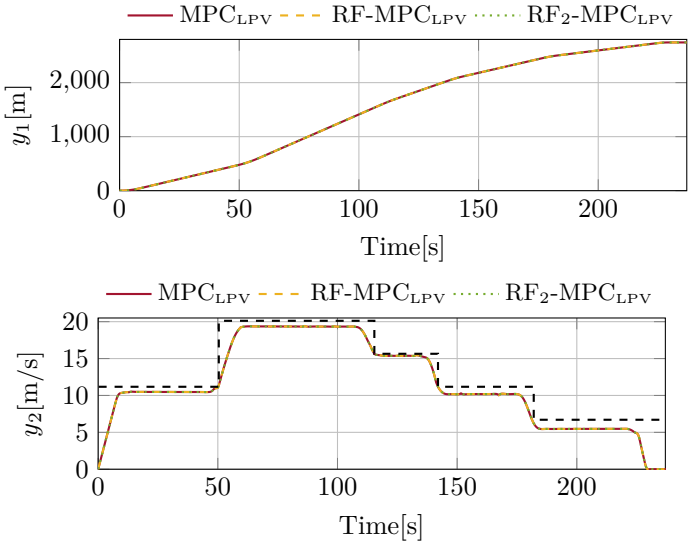

Figure 5.11: Comparison of output position y_1 and speed y_2 when driving the tram with MPC$_{\text{LPV}}$, RF-MPC$_{\text{LPV}}$and RF$_2$-MPC$_{\text{LPV}}$ algorithms together with the output constraints.

Chapter 6

Conclusions

Model Predictive Control (MPC) algorithms proposed in this thesis address different problems affecting electrical and mechatronic systems. Control systems have been tested on real processes and high fidelity simulators.

The proposed algorithms for power converters have been oriented to improve pre-compensated systems performance allowing to embed the considered algorithms on commercial hardware, maintaining the original low level controllers and limiting changes in the designed primal control loop. Theoretical conditions to effectively apply MPC for pre-compensed systems on low computational hardware have been also given.

Control laws developed for piezoelectric micro-positioning systems also represent a solution to embed these algorithms in commercial control hardware. Both MPC and RG paradigm have been considered in the development of the proposed algorithms, showing the ability of the predictive control to drive nonlinear plants considering approximated linear models of the real process.

Algorithms to control Unmanned Vehicles have been also considered. A Reference Governor (RG) approach for a VTOL aircraft has been proposed and a Fault Tolerant (FT) policy for an over-actuated vessel has been developed. Both solutions are based on Linear Time Invariant (LTI) MPC. The nonlinear behavior of the controlled systems has been approximated with the identified or linearized models, exploiting MPC robustness properties to control the considered processes.

MPC solutions to control Linear Parameter Varying (LPV) models have been investigated. Considering the anticipative action, several algorithms have been developed to improve the control performance of an Aerosonde aircraft described by an identified LPV model. An innovative algorithm oriented to reduced the computational effort required to compute the LPV-MPC control law has been also presented. By the introduction of a coordinate transformation in the control law, the developed algorithm avoids to iteratively re-compute the MPC optimization problem, reducing the required computational burden. This theoretical result allows to reduce the computational effort of LPV-based MPC on embedded control platform.

Several research opportunities have been highlighted by the application of the LPV paradigm to complex nonlinear dynamics plants. Therefore the future research will be oriented to the study of alternative solutions to the standard LPV-MPC algorithms, in order to the improve the computational efficiency and investigating the features related to the iterative rebuild of the control law. In particular, due to the LPV-MPC paradigm, the possibility to change the control law during the rebuilding step, modifying it with respect to the control scenario, opens the way to a wide set of research opportunities.

Bibliography

[1] Ion Necoara, "Computational complexity certification for dual gradient method: Application to embedded mpc," *Systems & Control Letters*, vol. 81, pp. 49–56, 2015.

[2] Marshall D Rafal and William F Stevens, "Discrete dynamic optimization applied to on-line optimal control," *AIChE journal*, vol. 14, no. 1, pp. 85–91, 1968.

[3] Ernest Bruce Lee and Lawrence Markus, "Foundations of optimal control theory," Tech. Rep., MINNESOTA UNIV MINNEAPOLIS CENTER FOR CONTROL SCIENCES, 1967.

[4] Eduardo F Camacho and Carlos Bordons Alba, *Model predictive control*, Springer Science & Business Media, 2013.

[5] Jan Marian Maciejowski, *Predictive control: with constraints*, Pearson education, 2002.

[6] David Q Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[7] Yang Wang and Stephen Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[8] Stefano Di Cairano, Hongtei Eric Tseng, Daniele Bernardini, and Alberto Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1236–1248, 2013.

[9] Luigi Del Re, Frank Allgöwer, Luigi Glielmo, Carlos Guardiola, and Ilya Kolmanovsky, "Automotive model predictive control," *Lecture Notes in Control and Information Science*, 2010.

[10] Stefano Di Cairano, Diana Yanakiev, Alberto Bemporad, Ilya V Kolmanovsky, and Davor Hrovat, "Model predictive idle speed control: Design, analysis, and experimental evaluation," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 1, pp. 84–97, 2012.

[11] Edward N Hartley and Jan M Maciejowski, "Field programmable gate array based predictive control system for spacecraft rendezvous in elliptical orbits," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 585–607, 2015.

[12] Sergio Vazquez, Jose I Leon, Leopoldo G Franquelo, Jose Rodriguez, Hector A Young, Abraham Marquez, and Pericle Zanchetta, "Model predictive control: A review of its applications in power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 16–31, 2014.

[13] Alberto Bemporad, "Reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 415–419, 1998.

[14] E. Garone and M.M. Nicotra, "Explicit Reference Governor for Constrained Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2015.

[15] So-Ryeok Oh and S.K. Agrawal, "A reference governor-based controller for a cable robot under input constraints," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 639–645, Jul 2005.

[16] Emanuele Garone, Stefano Di Cairano, and Ilya Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[17] Shyam Jade, Erik Hellström, Jacob Larimore, Anna G Stefanopoulou, and Li Jiang, "Reference governor for load control in a multicylinder recompression hcci engine," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1408–1421, 2014.

[18] Luca Cavanini, Gionata Cimini, and Gianluca Ippoliti, "Model predictive control for the reference regulation of current mode controlled dc-dc converters," in *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*. IEEE, 2016, pp. 74–79.

[19] Luca Cavanini, Gionata Cimini, Gianluca Ippoliti, and Alberto Bemporad, "Model predictive control for pre-compensated voltage mode controlled dc-dc converters," *IET Control Theory & Applications*, 2017.

[20] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*. Kobe, 2009, vol. 3, p. 5.

[21] Ralph Tyrell Rockafellar, *Convex analysis*, Princeton university press, 2015.

[22] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[23] Petter TøNdel, Tor Arne Johansen, and Alberto Bemporad, "An algorithm for multi-parametric quadratic programming and explicit mpc solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[24] Mayuresh V Kothare, Vesna Nevistic, and Manfred Morari, "Robust constrained model predictive control for nonlinear systems: a comparative study," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*. IEEE, 1995, vol. 3, pp. 2884–2885.

[25] Alberto Bemporad and Carlo Filippi, "Suboptimal explicit mpc via approximate multiparametric quadratic programming," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*. IEEE, 2001, vol. 5, pp. 4851–4856.

[26] S Joe Qin and Thomas A Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[27] Rudolph Emil Kalman et al., "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[28] Jeff S Shamma, *Analysis and design of gain scheduled control systems*, Ph.D. thesis, Massachusetts Institute of Technology, 1988.

[29] Jeff S Shamma, "An overview of lpv systems," in *Control of linear parameter varying systems with applications*, pp. 3–26. Springer, 2012.

[30] Wilson J. Rugh and Jeff S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, oct 2000.

[31] Bart Paijmans, Wim Symens, Hendrik Van Brussel, and Jan Swevers, "Identification of interpolating affine lpv models for mechatronic systems with one varying parameter," *European Journal of Control*, vol. 14, no. 1, pp. 16–29, 2008.

[32] Carsten W Scherer, "Lpv control and full block multipliers," *Automatica*, vol. 37, no. 3, pp. 361–375, 2001.

[33] Bassam Bamieh and Laura Giarre, "Identification of linear parameter varying models," *International journal of robust and nonlinear control*, vol. 12, no. 9, pp. 841–853, 2002.

[34] Francesco Casella and Marco Lovera, "Lpv/lft modelling and identification: overview, synergies and a case study," in *Computer-Aided Control Systems, 2008. CACSD 2008. IEEE International Conference on*. IEEE, 2008, pp. 852–857.

[35] J Abonyi and R Babuska, "Local and global identification and interpretation of parameters in takagi-sugeno fuzzy models," in *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*. IEEE, 2000, vol. 2, pp. 835–840.

[36] Amol A Khalate, Xavier Bombois, Roland Tóth, and Robert Babuŝka, "Optimal experimental design for lpv identification using a local approach," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 162–167, 2009.

[37] R.J. Hesseling, M. Steinbuch, F.E. Veldpaus, and T. Klisch, "Identification and control for future restraint systems," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*. IEEE.

[38] Bart Paijmans, Wim Symens, Hendrik Van Brussel, and Jan Swevers, "Identification of interpolating affine LPV models for mechatronic systems with one varying parameter," *European Journal of Control*, vol. 14, no. 1, pp. 16–29, jan 2008.

[39] Marco Lovera and Guillaume Mercere, "Identification for gain-scheduling: a balanced subspace approach," in *2007 American Control Conference*. jul 2007, IEEE.

[40] Yucai Zhu and Zuhua Xu, "A method of LPV model identification for control," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 5018–5023, 2008.

[41] T. Gustafsson, M. Lovera, and M. Verhaegen, "A novel algorithm for recursive instrumental variable based subspace identification," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*. IEEE.

[42] Kameshwar Poolla, "Identification of linear parameter-varying systems using nonlinear programming^," *Journal of Dynamic Systems, Measurement and Control*, vol. 121, pp. 71–78, 1999.

[43] Jan-Willem van Wingerden and Michel Verhaegen, "Subspace IDentification of MIMO LPV systems: The PBSID approach," in *2008 47th IEEE Conference on Decision and Control*. 2008, IEEE.

[44] Hossam S Abbas, Roland Tóth, Nader Meskin, Javad Mohammadpour, and Jurre Hanema, "An MPC approach for lpv systems in input-output form," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 91–96.

[45] Thomas Besselmann, Johan Löfberg, and Manfred Morari, "Explicit MPC for LPV systems: Stability and optimality," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2322–2332, 2012.

[46] Alberto Bemporad, Alessandro Casavola, and Edoardo Mosca, "A predictive reference governor for constrained control systems," *Computers in industry*, vol. 36, no. 1, pp. 55–64, 1998.

[47] Ilya Kolmanovsky, Emanuele Garone, and Stefano Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *2014 American Control Conference*. jun 2014, IEEE.

[48] Ilya Kolmanovsky, Emanuele Garone, and Stefano Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 226–241.

[49] Emanuele Garone, Francesco Tedesco, and Alessandro Casavola, "Distributed coordination-by-constraint strategies for networked control systems," *IFAC Proceedings Volumes*, vol. 42, no. 20, pp. 144–149, 2009.

[50] Uroš Kalabić, Ilya Kolmanovsky, Julia Buckland, and Elmer Gilbert, "Reduced order reference governor," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 3245–3251.

[51] Stefano Di Cairano and Ilya V Kolmanovsky, "Rate limited reference governor for network controlled systems," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 3704–3709.

[52] Dario Piga, Simone Formentin, and Alberto Bemporad, "Direct data-driven control of constrained systems," *IEEE Transactions on Control Systems Technology*, 2017.

[53] G. Cimini, G. Ippoliti, G. Orlando, and M. Pirro, "Sensorless power factor control for mixed conduction mode boost converter using passivity-based control," *IET Power Electronics*, vol. 7, no. 12, pp. 2988–2995, 2014.

[54] C. Bordons and C. Montero, "Basic Principles of MPC for Power Converters: Bridging the Gap Between Theory and Practice," *IEEE Industrial Electronics Magazine*, vol. 9, no. 3, pp. 31–43, Sept 2015.

[55] Maher Algreer, Matthew Armstrong, and Damian Giaouris, "Adaptive PD+ I control of a switch-mode DC–DC power converter using a recursive FIR predictor," *IEEE Transactions on Industry Applications*, vol. 47, no. 5, pp. 2135–2144, 2011.

[56] G. Cimini and A. Bemporad, "Exact Complexity Certification of Active-Set Methods for Quadratic Programming," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2017.

[57] Z. Zhang, F. Wang, T. Sun, J.R. Rodriguez, and R. Kennel, "FPGA Based Experimental Investigation of a Quasi-Centralized DMPC Scheme for a Back-to-Back Converter," *IEEE Transactions on Power Electronics*, vol. PP, no. 99, pp. 1–1, 2015.

[58] J. Rodriguez, M.P. Kazmierkowski, J.R. Espinoza, P. Zanchetta, H. Abu-Rub, H.A. Young, and C.A. Rojas, "State of the Art of Finite Control Set Model Predictive Control in Power Electronics," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 1003–1016, May 2013.

[59] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3 – 20, 2002.

[60] S. K. Kim, C. R. Park, J. S. Kim, and Y. I. Lee, "A Stabilizing Model Predictive Controller for Voltage Regulation of a DC/DC Boost Converter," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 2016–2023, Sept 2014.

[61] R.P. Aguilera, P. Lezana, and D.E. Quevedo, "Switched Model Predictive Control for Improved Transient and Steady-State Performance," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 968–977, Aug 2015.

[62] F. Kurokawa, A. Yamanishi, and S. Hirotaki, "A Reference Modification Model Digitally Controlled DC-DC Converter for Improvement of Transient Response," *IEEE Transactions on Power Electronics*, vol. 31, no. 1, pp. 871–883, Jan 2016.

[63] Kiminao Kogiso and Kenji Hirata, "Reference governor for constrained systems with time-varying references," *Robotics and Autonomous Systems*, vol. 57, no. 3, pp. 289 – 295, 2009.

[64] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *American Control Conference (ACC), 2014*, June 2014, pp. 226–241.

[65] R.W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, Norwell, MA: Kluwer, 2001.

[66] H. Sira-Ramirez and R. Silva-Ortigoza, *Control Design Techniques in Power Electronics Devices*, Power Systems. Springer London, 2006.

[67] S. Mariethoz, S. Almer, M. Baja, A.G. Beccuti, D. Patino, A. Wernrud, J. Buisson, H. Cormerais, T. Geyer, Hisaya Fujioka, U.T. Jonsson, Chung-Yao Kao, M. Morari, G. Papafotiou, A. Rantzer, and P. Riedinger, "Comparison of Hybrid Control Techniques for Buck and Boost DC-DC Converters," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1126–1145, Sept 2010.

[68] R.B. Ridley, "A new, continuous-time model for current-mode control power converters," *IEEE Transactions on Power Electronics*, vol. 6, no. 2, pp. 271–280, Apr 1991.

[69] Texas Instrument, "PTD08A010WAD Datasheet," `https://www.ti.com/lit/ds/symlink/ptd08a010w.pdf`, 2007, [Online; Revisied Febraury 2010].

[70] Guo-Ying Gu, Li-Min Zhu, Chun-Yi Su, Han Ding, and Sergej Fatikow, "Modeling and control of piezo-actuated nanopositioning stages: a survey," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 313–332, 2016.

[71] YK Yong, SO Reza Moheimani, B J_ Kenton, and KK Leang, "Invited review article: High-speed flexure-guided nanopositioning: Mechanical design and control issues," *Review of scientific instruments*, vol. 83, no. 12, pp. 121101, 2012.

[72] Tomas Tuma, Abu Sebastian, John Lygeros, and Angeliki Pantazi, "The four pillars of nanopositioning for scanning probe microscopy: The position sensor, the scanning device, the feedback controller, and the reference trajectory," *IEEE Control Systems*, vol. 33, no. 6, pp. 68–85, 2013.

[73] Mei-Ju Yang, Guo-Ying Gu, and Li-Min Zhu, "High-bandwidth tracking control of piezo-actuated nanopositioning stages using closed-loop input shaper," *Mechatronics*, vol. 24, no. 6, pp. 724–733, 2014.

[74] Yingfeng Shan and Kam K Leang, "Dual-stage repetitive control with prandtl–ishlinskii hysteresis inversion for piezo-based nanopositioning," *Mechatronics*, vol. 22, no. 3, pp. 271–281, 2012.

[75] Lei Liu, Yanbin Zhao, Liang Tang, Yufei Xu, Yu-guang Bai, and He Liao, "Modeling and identification investigation of multi-field hysteretic dynamics in flexure-guided piezo platform," *Mechanical Systems and Signal Processing*, vol. 50, pp. 594–606, 2015.

[76] Henry Frank Tiersten, *Linear Piezoelectric Plate Vibrations: Elements of the Linear Theory of Piezoelectricity and the Vibrations Piezoelectric Plates*, Springer, 2013.

[77] Santosh Devasia, Evangelos Eleftheriou, and SO Reza Moheimani, "A survey of control issues in nanopositioning," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 5, pp. 802–823, 2007.

[78] Jack W Macki, Paolo Nistri, and Pietro Zecca, "Mathematical models for hysteresis," *SIAM review*, vol. 35, no. 1, pp. 94–123, 1993.

[79] Mark A Krasnosel'skii and Aleksei V Pokrovskii, *Systems with hysteresis*, Springer Science & Business Media, 2012.

[80] Yanfang Liu, Jinjun Shan, Ulrich Gabbert, and Naiming Qi, "Hysteresis and creep modeling and compensation for a piezoelectric actuator using a fractional-order maxwell resistive capacitor approach," *Smart Materials and Structures*, vol. 22, no. 11, pp. 115020, 2013.

[81] Luca Cavanini, Maria Letizia Corradini, Luigino Criante, Andrea Di Donato, Marco Farina, Gianluca Ippoliti, Sara Lo Turco, Giuseppe Orlando, and Carmine Travaglini, "Robust control of piezostage for nanoscale three-dimensional images acquisition," in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. IEEE, 2016, pp. 5107–5112.

[82] Peng Chao, Chongwei Han, Jianxiao Zou, and Guanghui Zhang, "H [infinity] optimal inversion feedforward and robust feedback based 2dof control approach for high speed-precision positioning systems," *Journal of Control Science and Engineering*, 2016.

[83] Qingsong Xu and Minping Jia, "Model reference adaptive control with perturbation estimation for a micropositioning system," *IEEE Transactions on control systems technology*, vol. 22, no. 1, pp. 352–359, 2014.

[84] Arnfinn Aas Eielsen, Mernout Burger, Jan Tommy Gravdahl, and Kristin Y Pettersen, "Pi 2-controller applied to a piezoelectric nanopositioner using conditional integrators and optimal tuning," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 887–892, 2011.

[85] Y Cao, L Cheng, XB Chen, and JY Peng, "An inversion-based model predictive control with an integral-of-error state variable for piezoelectric actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 895–904, 2013.

[86] Qingsong Xu, "Digital sliding mode prediction control of piezoelectric micro/nanopositioning system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 297–304, 2015.

[87] Md Sohel Rana, Hemanshu R Pota, and Ian R Petersen, "The design of model predictive control for an afm and its impact on piezo nonlinearities," *European Journal of Control*, vol. 20, no. 4, pp. 188–198, 2014.

[88] Vijay A Neelakantan, Gregory N Washington, and Norman K Bucknor, "Model predictive control of a two stage actuation system using piezoelectric actuators for controllable industrial and automotive brakes and clutches," *Journal of Intelligent Material Systems and Structures*, vol. 19, no. 7, pp. 845–857, 2008.

[89] Dong Zhang, Cheng-Jin Zhang, Qiang Wei, Yan-Bing Tian, Jing-Bo Zhao, and Xian-Ming Li, "Modeling and control of piezo-stage using neural networks," *Guangxue Jingmi Gongcheng(Optics and Precision Engineering)*, vol. 20, no. 3, pp. 587–596, 2012.

[90] Michel Verhaegen and Paul Van Dooren, "Numerical aspects of different kalman filter implementations," *IEEE Transactions on Automatic Control*, vol. 31, no. 10, pp. 907–917, 1986.

[91] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 65–100, nov 2009.

[92] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *AIAA Guidance, Navigation and Control Conference and Exhibit*. aug 2007, American Institute of Aeronautics and Astronautics.

[93] Paul Bevilaqua, Richard Margason, and Charles Gaharan, "Jet entrainment theory for vertical takeoff and landing aircraft suckdown," *AIAA Journal*, vol. 48, no. 2, pp. 330–339, feb 2010.

[94] Bryan Brown, Wei Wei, Rodger Ozburn, Manish Kumar, and Kelly Cohen, "Surveillance for intelligent emergency response robotic aircraft (SIERRA)- VTOL aircraft for emergency response," in *AIAA Infotech @ Aerospace*. jan 2015, American Institute of Aeronautics and Astronautics.

[95] Patrick C. Murphy and Drew Landman, "Experiment design for complex VTOL aircraft with distributed propulsion and tilt wing," in *AIAA Atmospheric Flight Mechanics Conference*. jan 2015, American Institute of Aeronautics and Astronautics.

[96] Luca Cavanini, Gionata Cimini, and Gianluca Ippoliti, "A fast model predictive control algorithm for linear parameter varying systems with right invertible input matrix," in *2017 25th Mediterranean Conference on Control and Automation (MED)*. jul 2017, IEEE.

[97] Dongkyoung Chwa, "Fuzzy adaptive output feedback tracking control of VTOL aircraft with uncertain input coupling and input-dependent disturbances," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1505–1518, oct 2015.

[98] Kristoffer Gryte, "High angle of attack landing of an unmanned aerial vehicle," M.S. thesis, NTNU, 2015.

[99] U. Dynamics, "Aerosim toolbox," www.u-dynamics.com, available: (2009, July).

[100] Jinhyun Kim, Min-Sung Kang, and Sangdeok Park, "Accurate modeling and robust hovering control for a quad–rotor VTOL aircraft," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 9–26, sep 2009.

[101] Tor A. Johansen and Thor I. Fossen, "Control allocation—a survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, may 2013.

[102] Jan-Willem van Wingerden and Michel Verhaegen, "Subspace identification of bilinear and LPV systems for open- and closed-loop data," *Automatica*, vol. 45, no. 2, pp. 372–381, feb 2009.

[103] Sarah C Rutan, "Adaptive kalman filtering," *Analytical Chemistry*, vol. 63, no. 22, pp. 1103A–1109A, 1991.

[104] Youmin Zhang and Jin Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, dec 2008.

[105] Ron J Patton, "Fault-tolerant control," *Encyclopedia of systems and control*, pp. 422–428, 2015.

[106] Shen Yin, Hao Luo, and Steven X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2402–2411, may 2014.

[107] Jin Jiang and Xiang Yu, "Fault-tolerant control systems: A comparative study between active and passive approaches," *Annual Reviews in Control*, vol. 36, no. 1, pp. 60–72, apr 2012.

[108] Adolfo J. Sanchez, Juan M. Escano, Antonio J. Gallego, and Eduardo F. Camacho, "Fault tolerant MPC of a solar trough field based on classification and regression trees," in *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*. sep 2016, IEEE.

[109] Tao Peng, Hanbing Dan, Jian Yang, Hui Deng, Qi Zhu, Chunsheng Wang, Weihua Gui, and Josep M. Guerrero, "Open-switch fault diagnosis and fault tolerant for matrix converter with finite control set-model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 9, pp. 5953–5963, sep 2016.

[110] Roger Skjetne, Øyvind Smogeli, and Thor I. Fossen, "Modeling, identification, and adaptive maneuvering of CyberShip II: A complete design with experiments," *IFAC Proceedings Volumes*, vol. 37, no. 10, pp. 203–208, jul 2004.

[111] Defeng Wu, Fengkun Ren, and Weidong Zhang, "An energy optimal thrust allocation method for the marine dynamic positioning system based on adaptive hybrid artificial bee colony algorithm," *Ocean Engineering*, vol. 118, pp. 216–226, may 2016.

[112] Carlos Henrique Farias dos Santos, Daisy Isabel Kang Cardozo, Romeu Reginatto, and Edson Roberto De Pieri, "Bank of controllers and virtual thrusters for fault-tolerant control of autonomous underwater vehicles," *Ocean Engineering*, vol. 121, pp. 210–223, jul 2016.

[113] Xiaoxu Liu, Zhiwei Gao, and Michael Z. Q. Chen, "Takagi–sugeno fuzzy model based fault estimation and signal compensation with application to wind turbines," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5678–5689, jul 2017.

[114] Zhiwei Gao, Xiaoxu Liu, and Michael Chen, "Unknown input observer based robust fault estimation for systems corrupted by partially-decoupled disturbances," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2015.

[115] Thor I Fossen, *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*, 2002.

[116] Thor I Fossen and Jann Peter Strand, "Passive nonlinear observer design for ships using lyapunov methods: full-scale experiments with a supply vessel," *Automatica*, vol. 35, no. 1, pp. 3–16, jan 1999.

[117] K.-P. Lindegaard and T.I. Fossen, "Fuel-efficient rudder and propeller control allocation for marine craft: experiments with a model ship," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 850–862, nov 2003.

[118] Thor I. Fossen and Tor A. Johansen, "A survey of control allocation methods for ships and underwater vehicles," in *2006 14th Mediterranean Conference on Control and Automation*. jun 2006, IEEE.

[119] Stanimir Mollov, Robert Babuska, Janos Abonyi, and Henk B Verbruggen, "Effective optimization for fuzzy model predictive control," *IEEE Transactions on fuzzy systems*, vol. 12, no. 5, pp. 661–675, 2004.

[120] Luca Cavanini, Luigi Colombo, Gianluca Ippoliti, and Giuseppe Orlando, "Development and experimental validation of a lqg control for a precompensated multi-axis piezosystem," in *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on*. IEEE, 2017, pp. 460–465.

[121] Daniele Corona and Bart De Schutter, "Adaptive cruise control for a SMART car: A comparison benchmark for MPC-PWA control methods," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 365–372, 2008.

[122] Luca Cavanini, Gionata Cimini, and Gianluca Ippoliti, "A fast model predictive control algorithm for linear parameter varying systems with right invertible input matrix," in *2017 25th Mediterranean Conference on Control and Automation (MED)*. jul 2017, IEEE.

[123] Chenguang Liu, Huarong Zheng, Rudy R Negenborn, Xiumin Chu, and Le Wang, "Trajectory tracking control for underactuated surface vessels based on nonlinear model predictive control," in *International Conference on Computational Logistics*. Springer, 2015, pp. 166–180.

[124] Huarong Zheng, Rudy R Negenborn, and Gabriel Lodewijks, "Trajectory tracking of autonomous vessels using model predictive control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 8812–8818, 2014.

[125] H Zheng, RR Negenborn, and G Lodewijks, "Model predictive control of a waterborne AGV at the operational level," in *Proceedings of the International Maritime and Port Technology and Development Conference (MTEC 2014)*, 2014, pp. 99–108.

[126] Tor A. Johansen and Thor I. Fossen, "Control allocation—a survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, may 2013.

[127] Zhiwei Gao, Carlo Cecati, and Steven X. Ding, "A survey of fault diagnosis and fault-tolerant techniques - part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, jun 2015.

[128] Andrea Cristofaro and Tor Arne Johansen, "Unknown input observers and fault-tolerant control allocation," Tech. Rep., Technical report, 2013.

[129] Andrea Cristofaro and Tor Arne Johansen, "Fault tolerant control allocation using unknown input observers," *Automatica*, vol. 50, no. 7, pp. 1891–1897, 2014.

[130] Xiaoxu Liu, Zhiwei Gao, Richard Binns, and Hui Shao, "Robust fault estimation for stochastic takagi-sugeno fuzzy systems," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. oct 2016, IEEE.

[131] Sarah Odofin, Zhiwei Gao, Xiaoxu Liu, and Kai Sun, "Robust actuator fault detection for an induction motor via genetic-algorithm optimisation," in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*. jun 2016, IEEE.

[132] Xiaoxu Liu and Zhiwei Gao, "Unknown input observers for fault diagnosis in lipschitz nonlinear systems," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*. aug 2015, IEEE.

[133] J Hanema, Roland Tóth, and M Lazar, "Tube-based anticipative linear parameter-varying mpc: application to non-linear systems," 2017.

[134] Aerosonde Pty Ltd, "Aerosonde," www.aerosonde.com.

[135] Herve Audren, Joris Vaillant, Abderrahmane Kheddar, Adrien Escande, Kenji Kaneko, and Eiichi Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. sep 2014, IEEE.

[136] J Hanema, Roland Tóth, M Lazar, and S Weiland, "Anticipative linear parameter-varying model predictive control," 2015.

[137] G. J. Holland, P. J. Webster, J. A. Curry, G. Tyrell, D. Gauntlett, G. Brett, J. Becker, R. Hoag, and W. Vaglienti, "The aerosonde robotic aircraft: A new paradigm for environmental observations," *Bulletin of the American Meteorological Society*, vol. 82, no. 5, pp. 889–901, may 2001.

[138] Peter Thomasson, "Aircraft control and simulation: Dynamics, controls design, and autonomous systems – third edition b. l. stevenset al. john wiley and sons, the atrium, southern gate, chichester, west sussex, UK, PO19 8sq. 2016. 749pp. illustrated. £110. ISBN 978-1-118-87098-3.," *The Aeronautical Journal*, vol. 121, no. 1236, pp. 269–270, feb 2017.

[139] Francois Bateman, Hassan Noura, and Mustapha Ouladsine, "Fault diagnosis and fault-tolerant control strategy for the aerosonde UAV," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2119–2137, jul 2011.

[140] D.M.K.K. Venkateswara Rao and Tiauw Hiong Go, "Automatic landing system design using sliding mode control," *Aerospace Science and Technology*, vol. 32, no. 1, pp. 180–187, jan 2014.

[141] Lalo Magni, Riccardo Scattolini, and Mara Tanelli, "Switched model predictive control for performance enhancement," *International Journal of Control*, vol. 81, no. 12, pp. 1859–1869, dec 2008.

[142] S. Di Cairano, H.E. Tseng, D. Bernardini, and Alberto Bemporad, "Steering vehicle control by switched model predictive control," *IFAC Proceedings Volumes*, vol. 43, no. 7, pp. 1–6, jul 2010.

[143] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," vol. 59, no. 1, pp. 18–33, 2014.

[144] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, Society for Industrial and Applied Mathematics, 1994.

[145] G. Cimini, D. Bernardini, A. Bemporad, and S. Levijoki, "Online model predictive torque control for permanent magnet synchronous motors," in *Industrial Technology (ICIT), 2015 IEEE International Conference on*, March 2015, pp. 2308–2313.

[146] A. Bemporad, "A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1111–1116, April 2016.

[147] Michael J. Kurtz and Michael A. Henson, "Feedback linearizing control of discrete-time nonlinear systems with input constraints," *International Journal of Control*, vol. 70, no. 4, pp. 603–616, 1998.

[148] D. Simon, J. Lofberg, and T. Glad, "Nonlinear model predictive control using feedback linearization and local inner convex constraint approximations," in *Control Conference (ECC), 2013 European*, July 2013, pp. 2056–2061.

[149] Nael H El-Farra, Antonios Armaou, and Panagiotis D Christofides, "Analysis and control of parabolic PDE systems with input constraints," *Automatica*, vol. 39, no. 4, pp. 715–725, 2003.

[150] Jiamei Deng, Victor Becerra, and Richard Stobart, "Input constraints handling in an MPC/feedback linearization scheme," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 219–232, 2009.

[151] Gionata Cimini, Youngki Kim, Buz McCain, Jason Siegel, and Anna Stefanopoulou, "Model predictive control for real-time position tracking of a catenary-free tram," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1000–1005, 2017.