Real-time optimal control of a low consumption electric vehicle.

Dolly Tatiana Manrique Espindola

Université de Lorraine

Département de formation doctorale en automatique

École doctorale IAEM Lorraine

Commande optimale d'une voiture électrique à faible consommation sous contraintes temps réel

THÈSE

présentée et soutenue publiquement le 09 Décembre 2014

pour l'obtention du

Doctorat de l'Université de Lorraine

Spécialité Automatique, Traitement du Signal et des Images, Génie Informatique

par

Dolly Tatiana MANRIQUE ESPINDOLA

Composition du jury

Président :	Gérard BLOCH	Professeur, CRAN Université de Lorraine, Nancy
Rapporteurs :	Teodoro ALAMO	Professeur, ETSI Universidad de Sevilla, Espagne
	Yann CHAMAILLARD	Professeur, LME Polytech'Orléans, Orléans
Examinateur :	Mario SIGALOTTI	Chargé de Recherche INRIA, CMAP Ecole Polytechnique, Saclay
Directeur :	Gilles MILLERIOUX	Professeur, CRAN Université de Lorraine, Nancy
Co-directeur :	Thomas CHAMBRION	Maître de conférences HDR, IECL Université de Lorraine, Nancy

Centre de Recherche en Automatique de Nancy
UMR 7039

Institute Élie Cartan de Lorraine
UMR 7502

Acknowledgements

I would like to express my special appreciation and thanks to all those persons who, in one way or another, supported me during the realization of this doctoral thesis.

In particular, i would like to thank my Ph.D advisors Professor Dr. Gilles Millerioux and Dr. Thomas Chambrion HDR for the encouraging and supporting guidelines given to me for the last three years. Every one of which allowed me to accomplish successfully this outstanding achievement.

I would also like to thank the members of the jury committee, Professor Dr. Gérard Bloch, Professor Dr. Teodoro Alamo, Professor Dr. Yann Chamaillard and Dr. Mario Sigalotti for accepting to be part of the committee and for allowing my defense to be an enjoyable and unforgettable moment.

I would like to thank in a very special way Dr. Mirko Fiacchini for sharing with me his knowledge and expertise, for his extremely useful advices, and also for his patience.

I would like to thank especially the engineers Eng. Hugo Malaise, Eng. Vincent Scharff and Eng. Laurent Martignon for all the invaluable technical support.

I would also like to thank the teachers members of the EcoMotion Team, Gérard Dechenaud, Jean-Pierre Sarteaux, Philippe Reb, Joël Landier, Pascal Jean and Jean-Claude Sivault, who kindly welcomed me from the very first time i joined the team.

I would like to thank Team leaders David Gousse, Sandra Rossi, Mathieu Chevalier, Mathieu Rupp, Dany Sacleux and Loïc Kern, and all

the students members of the staff of the EcoMotion Team since the year 2011, for their collaboration in the culmination of this work.

I would like to give exceptional thanks to the teachers of CRAN and members of ESSTIN, Floriane Collin, Marion Gilson, Jean-Luc Noizette, Eric Gnaedinger, Vincent Laurain, Gaëtan Didier, Hugues Garnier, Gérard Bloch, Sylvie Ferrari, Marie-Christine Suhner, Philippe Weber and Frédéric Sarry, who fostered me from the very first day of my arrival to the ESSTIN, and who shared with me priceless lunch and coffee-times.

I would like to thank also my Ph.D colleagues and ex-Ph.D students, Dr. Jérémy Parriaux, Dr. Meriem Halimi, Dr. Fengwei Chen, Dr. Julien Schorsch, Dr. Boyi Ni, Dr. Mahmoud Abdelrahim and (futures) Dr. Brandon Dravie, Dr. Artuto Padilla and Dr. Yusuf Bhujwalla, which i had the very good fortune to meet at ESSTIN.

I would like also to thank the Dr. Francisco Lopez Estrada and the also future doctors Dr. Carolina Abscheidt and Dr. Eber Espino for the very nice moments *entre bastidores*.

Finally, I would like to thank my parents Hernán and Cecilia for their support, their prayers and love from that far away. My baby sister and my baby brother Laudy and Manuelito, for their unconditional love. I like to thank Henry Carrillo for his invaluable support and without whom anything of this would be even possible.

I thank you all (and all who i may have forgotten to mention). You all must know that every one of you was crucial in the success of this work.

Résumé

Formulation du probléme

La recherche de la meilleure utilisation de l'énergie pour un véhicule donné revient à déterminer comment ce véhicule doit être conduit, de façon à minimiser la quantitée d'énergie utilisée pour un trajet donné en un temps donné. Pour un véhicule automobile évoluant dans le trafic, par nature changeant et imprévisible, le recherche d'une telle stratégie de conduite implique la prise en compte de contraintes en temps réel.

Cette thèse est principalement consacrée à ce problème, aussi bien dans les aspects théoriques que pratiques, pour le cas de *Vir'Volt* (Figure 1), un prototype électrique participant à des compétitions de véhicules à basse consommation. Particulièrement, le prototype *Vir'Volt* participe à l'European Shell Eco-Marathon, course européenne où le vainqueur aura parcouru une distance donnée à une vitesse moyenne imposée en utilisant la quantité minimale d'énergie. Plusieurs

Figure 1: Le *Vir'Volt* prototype lors de l'European Shell Eco-Marathon 2011.

approches sont successivement étudiées dans la solution de ce problème.

Calcul hors-ligne de la stratégie de conduite

En première approche, on considère un modèle linéaire en temps discret. Ce modèle est obtenu par identification expérimentale (Figure 2). On déduit en-

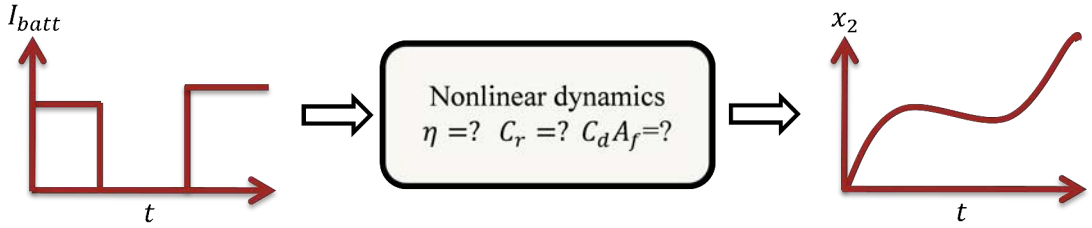


Figure 2: Identification expérimentale.

suite du modèle, après optimisation hors-ligne, une stratégie de conduite à faible consommation correspondant à un parcours optimal (stratégie de conduite optimale). Une Commande Prédictive (MPC) est implémentée dans le véhicule pour suivre, en temps réel, cette trajectoire. La commande prédictive prend en compte l'état réel du véhicule, et l'écart à la position idéale, ainsi que des contraintes imposées en temps réel (correspondant par exemple à des limitations de vitesse en entrée de virages ou lors de densification du trafic). De manière très classique, la MPC proposée consiste à calculer, à chaque pas de temps, une commande permettant d'amener le système dans un ensemble invariant (ici: un polytope) où une commande par retour d'état linéaire garantira la stabilité (Figure 3).

Transformation homothétique de l'ensemble invariant

Du fait du caractère variable des contraintes, un tel ensemble invariant varie et doit être recalculé à chaque pas de temps. La principale nouveauté consiste en un artifice destiné à alléger considérablement la charge de calcul: l'ensemble invariant retenu sera l'image par une homothétie (de rapport variable) d'un ensemble invariant fixe calculé hors-ligne (Figure 4).

Le calcul de la commande revient essentiellement à déterminer, à chaque pas de temps, le rapport de cette homothétie. L'algorithme qu'on en déduit est suffisamment léger pour être implémenté sur un micro-contrôleur embarqué. Sur un

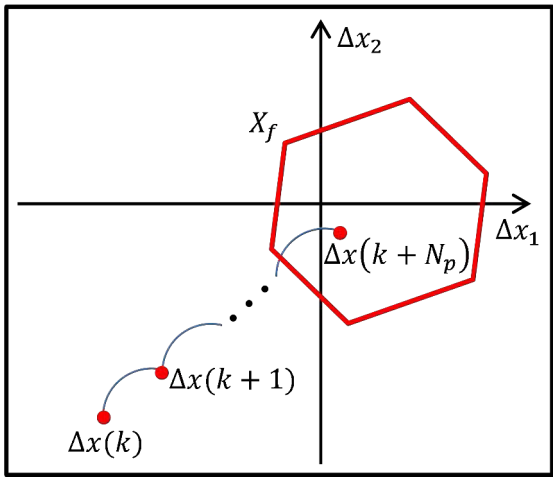


Figure 3: Ensemble invariant polytopique X_f.

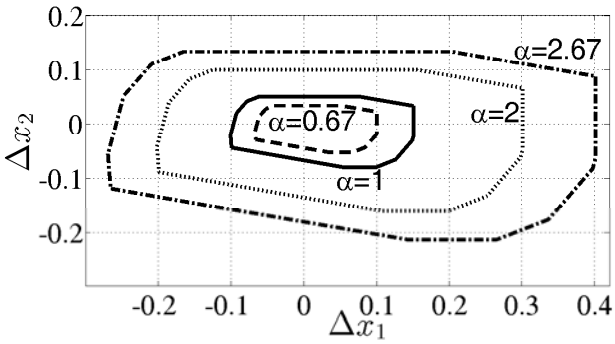


Figure 4: Le nouvel ensemble invariant est l'image par une homothétie d'un ensemble invariant fixe calculé hors-ligne.

plan théorique, on montre la stabilité et la convergence de la commande prédictive variant dans temps.

MPC pour systèmes Linéaires à Paramètre Variable (LPV) avec des ensembles invariants ellipsoïdales et Fonctions de Lyapunov Dépendantes de Paramètre (PDLF)

Dans une deuxième étape, pour mieux prendre en compte les caractéristiques non linéaires du système et profiter de la nature linéaire de la commande prédictive,

on considère un modèle LPV (Linéaire à Paramètre Variable) en temps discret. Différentes approches de construction d'une MPC sont envisagées. Les méthodes de calcul de l'ensemble invariant ellipsoïdal, reposant sur des Inégalités Matricielles Linéaires (LMI), sont testées, en-ligne ou avec report partiel de calculs hors-ligne. Le meilleur compromis pour un calcul embarqué est atteint avec un calcul hors-ligne et l'utilisation d'une Fonction de Lyapunov Dépendant de Paramètre (PDLF). La forme explicite de la commande permet une analyse complète de sa stabilité malgré la non-linéarité du modèle. Les résultats expérimentaux (avec implémentation sur le prototype et calcul de la commande sur micro-contrôleur (Figure 5)) montrent de bonnes performances de suivi de trajectoire pour de petites perturbations.

Figure 5: Commande embarqée sur micro-contrôleur.

Calcul en-ligne de la stratégie de conduite

Les deux premières approches avaient pour défaut d'être des modèles en temps discret. Les moyens de calcul embarqué volontairement limités ne permettent que des calculs à horizon court. Mais une précision acceptable du modèle suppose des pas de temps souvent petits, et donc des horizons lointains dans les calculs de MPC pour garantir la stabilité. Le rendement très mauvais à bas couple impose des stratégies de commande de type *on-off*, avec la difficulté supplémentaire que chaque démarrage du moteur suppose de remettre en mouvement la chane

de transmission, ce qui entrane une petite mais non négligeable consommation d'énergie. Ce caractère non lisse du coût est en général très difficile à prendre en compte d'un point de vue numérique. On propose ici une commande adaptative, à horizon variable, basée sur une heuristique très simple consistant à faire osciller la vitesse du véhicule entre deux paliers de vitesse. L'horizon d'optimisation dépend de la dynamique, identifiée en temps réel. Le caractère adaptatif confère une grande stabilité. L'implémentation est possible pour un coût calculatoire extrêmement faible (la puissance moyenne consommée par le micro-contrôleur utilisé est de l'ordre de 10 mW). Cette méthode a été utilisée pour la commande entièrement automatique du véhicule (avec contrôle de la position par GPS (Figure 6)) lors de la compétition officielle en 2014, pour des performances énergétiques comparables aux résultats des pilotes humains.

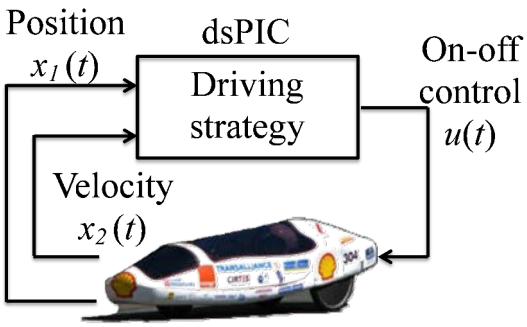


Figure 6: Commande entièrement automatique du véhicule lors du Shell Eco-Marathon 2014.

Abstract

The main objective of this thesis is to propose controlled driving strategies for an electric vehicle prototype that must achieve a minimal energetic consumption. The prototype under consideration is named *Vir'volt* and is involved every year in the Shell Eco-Marathon European race. The main issue, which can be considered as the guiding principle of this work, is to propose controlled driving strategies which can be embedded into the digital devices of the vehicle. As a result, the driving strategies must be compatible with real time constraints, limited memory and computational capacities of the electronic equipment. The computation of the strategies must require itself a low power consumption.

First, a suitable nonlinear model of the electric vehicle is obtained. The model involves physical equations with parameters estimated from experiments conducted on the vehicle. By suitable, it is meant a model fulfilling the trade-off complexity/precision needed for real-time control purposes.

The first overall approach consists in deriving, first, an optimal driving strategy by solving off-line an optimization problem. The problem amounts to minimizing a cost subject to constraints such as the dynamics of the vehicle, the physical constraints on the vehicle and the race (track profile and length, maximal duration of a run, etc.). As a second step, robust tracking methods of the optimal driving strategy are proposed. Mainly motivated by their robustness and constraint handling properties, several Model Predictive Control (MPC) tracking strategies are detailed.

A MPC tracking strategy based on a linearised model around an operating point is applied to the vehicle in simulation. Time-invariant constraints in the form of polytopic sets, are considered on the input and the state. The asymptotic stability of the control law is guaranteed by resorting to an invariant set as an admissible terminal constraint. Then, motivated by the peculiarities induced by

the tracking problem, time-varying constraints are considered, again in a polytopic form. The complexity of the tracking strategy is preserved, compared to the time-invariant case, by resorting to an homothetic transformation of a nominal invariant set, guaranteeing the asymptotic stability. The resulting MPC tracking strategy is assessed on the model of the *Vir'volt* vehicle in simulation.

To capture the nonlinearities of the dynamics, a Linear Parametric Varying (LPV) model is considered. A MPC strategy for LPV systems is proposed. The contribution that must be pointed out is that the approach is well suited for real-time applications, since it does not involve the on-line solution of any Linear Matrix Inequality (LMI) in the computation of the control law. The LMIs guarantee the stability and the constraints fulfilment. The performances of the approach, in terms of real-time applicability and robustness, are tested with success on the benchmark for the *Vir'volt* vehicle.

The principle of the second approach differs from the first one in the sense that the optimal driving strategy is computed on-line so that it can be adapted to a time-varying context. This is precisely the case when there is traffic jam during the race and when phenomena such as wind, rain and path irregularities are considered. The practical consideration that the efficiency of the power converter may not be optimal on all the operating range is also taken into account. This motivates an on-off strategy. The resulting on-off adaptive strategy requires an identification performed on-line of the model of the vehicle and of the disturbances. The robust adaptive control is embedded in a dsPIC device on-board of the *Vir'volt* vehicle, and is tested with success during the Shell Eco-Marathon 2014.

Publications

The publications related to this work are listed below:

- T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. MPC-based tracking for real-time systems subject to time-varying polytopic constraints. In *Optimal Control Applications and Methods (OCAM)*, John Wiley and Sons (2014) (To be published in 2015).
- T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. MPC tracking under time-varying polytopic constraints for real-time applications. In *European Control Conference (ECC)*, 2014. Strasbourg, France, 24 - 27 June 2014. DOI: 10.1109/ECC.2014.6862584.
- T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. MPC for a low-consumption electric vehicle with time-varying constraints. In *IFAC Joint Conference*, 2013. Grenoble, France, 04 - 06 February 2013. DOI: 10.3182/20130204-3-FR-2033.00213.
- T. Manrique, H Malaise, M. Fiacchini, T. Chambrion, and G. Millerioux. Model predictive real-time controller for a low-consumption electric vehicle. In *International Symposium on Environment-Friendly Energies and Applications IEEE (EFEA)*, 2012. Newcastle, UK, 25 - 27 June 2012. DOI: 10.1109/EFEA.2012.6294080. (Best paper award).

Contents

Contents xiii

List of Figures xix

List of Tables xxiii

Acronyms xxvii

1 General Introduction 1

1.1 The European Shell Eco-Marathon 2

1.1.1 Categories of participation 3

1.1.2 The Shell Eco-Marathon around the world 5

1.2 The EcoMotionTeam 6

1.3 Motivation of the work 9

1.4 Outline 9

Notation 1

2 The Low Consumption Vir'volt Electric Vehicle 11

2.1 Introduction 11

2.2 The Vir'volt prototype 12

2.2.1 Electric vehicle dynamics 12

2.2.2 Parameter identification 14

2.2.2.1 Nonlinear grey-box identification 15

2.2.2.2 Parameter estimation 15

2.3 Low consumption driving strategy 16

2.3.1 Energetic considerations . . . 17
2.3.2 Optimization problem . . . 19
2.3.2.1 Constraints of the Optimality problem . . . 19
2.3.2.2 The Multi-phase Optimality problem . . . 21
2.4 Nonlinear discrete-time model . . . 23
2.5 Real-time tracking of the optimal driving strategy . . . 25
2.5.1 Linearised model . . . 28
2.5.2 Linear Parametric Varying model . . . 29
2.6 Benchmark . . . 32
2.7 Conclusions . . . 37

3 Tracking Model Predictive Control 39
3.1 Introduction . . . 39
3.2 Preliminaries . . . 40
3.2.1 Polytopic constraints . . . 41
3.2.2 Problem formulation . . . 42
3.3 Design of the invariant terminal set . . . 45
3.4 MPC-based tracking with time-invariant constraints: an academic example . . . 48
3.4.1 Computation of the terminal invariant set . . . 50
3.4.2 Closed loop response . . . 52
3.5 MPC-based tracking with time-invariant constraints: application to the Vir'volt vehicle . . . 54
3.6 Conclusions . . . 57

4 Tracking under time-varying polytopic constraints 59
4.1 Introduction . . . 59
4.2 Preliminaries . . . 61
4.3 Homothetic transformation of the invariant set . . . 65
4.3.1 Principle of the homothetic transformation . . . 65
4.3.2 Computation of the homothetic factor . . . 66
4.4 MPC with homothetic transformation of the invariant set . . . 68

4.5 MPC-based tracking with time-varying constraints: an academic example . . . 69
4.5.1 Problem statement . . . 69
4.5.2 Results . . . 72
4.5.3 Computational resources . . . 76
4.6 MPC-based tracking with time-varying constraints: application to the Vir'volt vehicle . . . 79
4.6.1 Results . . . 81
4.7 Conclusions . . . 86

5 Real-time Robust Model Predictive Control for LPV systems 89
5.1 Introduction . . . 89
5.2 Problem formulation . . . 91
5.3 Robust constrained MPC for LPV systems . . . 93
5.3.1 Explicit MPC for LPV systems with off-line computation of LMIs . . . 97
5.3.1.1 Asymptotically stable invariant ellipsoids . . . 98
5.3.1.2 The explicit MPC algorithm . . . 99
5.4 Robust MPC for LPV systems: application to the Vir'volt vehicle 101
5.4.1 Robust MPC for LPV systems with on-line computation of LMIs . . . 102
5.4.2 Explicit MPC for LPV systems with off-line computation of LMIs . . . 104
5.4.3 Comparison of the fully on-line MPC and explicit MPC . . 107
5.5 MPC for LPV systems with Parameter dependent Lyapunov function 108
5.6 Explicit MPC using the PDLF . . . 112
5.6.1 The asymptotically stable invariant set . . . 112
5.6.2 Algorithm of the explicit MPC based on PDLF . . . 115
5.7 Robust MPC for LPV systems with PDLF: application to the Vir'volt vehicle . . . 117
5.7.1 Robust MPC with PDLF with on-line computation of LMIs 118
5.7.2 Explicit MPC for LPV using PDLF . . . 120

5.7.3 Comparison of fully on-line MPC based on PDLF and explicit MPC based on PDLF . . . 124
5.8 Explicit MPC for LPV using PDLF: application to the benchmark 126
5.9 Conclusions . . . 132

6 Robust adaptive real-time control based on an on-off driving srategy 133
6.1 Introduction . . . 133
6.2 Parameter identification . . . 135
6.2.1 Problem formulation . . . 136
6.2.2 Identification of parameter a . . . 138
6.2.2.1 Algorithm of the off-line identification . . . 140
6.2.3 Identification of parameters b and c . . . 142
6.3 Low consumption driving strategy . . . 144
6.3.1 Problem formulation . . . 144
6.3.2 Former results . . . 146
6.3.3 Periodic low consumption driving strategy . . . 147
6.4 Robust adaptive real-time control based on an on-off driving strategy: application to the Vir'volt vehicle . . . 154
6.4.1 Off-line identification of the parameter a . . . 154
6.4.2 On-line adaptative real-time control . . . 155
6.5 Conclusions . . . 159

7 General conclusions and Perspectives 161

A Rotterdam's Ahoy circuit 167

B Proofs of Chapter 5 169
B.1 Schur complement . . . 169
B.2 Proof of equation (5.27) . . . 169
B.3 Proof of equation (5.28) . . . 172
B.4 Proof of equation (5.31) . . . 174
B.5 Proof of equation (5.32) . . . 175

C Proofs of Chapter 6 **179**

C.1 Proof of equation (6.13) 179

References **183**

List of Figures

1.1 Shell Eco-Marathon Europe 2014 2
1.2 Rotterdam's Ahoy circuit 2014 3
1.3 Prototype and UrbanConcept vehicles 5
1.4 Prototypes developed by the EMT 6

2.1 The *Vir'Volt* prototype in the Shell Eco-Marathon 2011 11
2.2 Free-body diagram 13
2.3 *Vir'volt* Power train 14
2.4 Grey-box nonlinear identification 15
2.5 Grey-box identification of the nonlinear dynamics for a flat path . 17
2.6 Example of a track with straight lines and curves 19
2.7 Consecutive phases in the Rotterdam's Ahoy circuit 23
2.8 Optimal driving strategy 24
2.9 Time delay of the discretization process 26
2.10 Functions of the vertices of the polytopic LPV representation . . . 32
2.11 Benchmark 33
2.12 Benchmark in the year 2014 34
2.13 On-board implemented electronics 35
2.14 Optimal driving strategy for the vehicle in the benchmark 37

3.1 Set X_f of final states 44
3.2 Reference $\bar{x}_1(k)$ 49
3.3 Reference $\bar{x}_2(k)$ 49
3.4 Polytopic constraints 51
3.5 Backward iterative algorithm 51

3.6 Tracking of the reference $\bar{x}_1(k)$ . . . 52
3.7 Tracking of the reference $\bar{x}_2(k)$ . . . 52
3.8 Tracking error $\Delta x_1(k)$ . . . 53
3.9 Tracking error $\Delta x_2(k)$ . . . 53
3.10 Tracking error $\Delta u(k)$ . . . 53
3.11 Optimal control $u^*(k)$ and tracking constraints . . . 55
3.12 Maximal invariant set . . . 55
3.13 Closed-loop implementation of the MPC . . . 56
3.14 Tracking of the reference $x_2^*(k)$ . . . 56
3.15 Tracking error $\Delta x_2(k)$ . . . 56
3.16 Tracking error $\Delta u(k)$ . . . 57

4.1 Geometry of the homothetic transformation . . . 67
4.2 Invariant set $\hat{\Omega}$ . . . 72
4.3 Time evolution of $\alpha(k)$ . . . 73
4.4 Constraints for $k < 30$ . . . 74
4.5 Homothetic transformation for $k < 30$ . . . 74
4.6 Constraints for $30 \leq k < 90$ or $k \geq 140$ . . . 74
4.7 Homothetic transformation for $30 \leq k < 90$ or $k \geq 140$ . . . 75
4.8 Constraints for $90 \leq k < 140$ . . . 75
4.9 Homothetic transformation for $90 \leq k < 140$ . . . 75
4.10 Dilation and contraction of the invariant set . . . 76
4.11 Tracking of the reference $\bar{x}_1(k)$ . . . 76
4.12 Tracking of the reference $\bar{x}_2(k)$ . . . 77
4.13 Tracking error $\Delta x_1(k)$ . . . 77
4.14 Tracking error $\Delta x_2(k)$ . . . 77
4.15 Tracking error $\Delta u(k)$ . . . 78
4.16 Time-varying tracking constraints $U_\Delta(k)$ . . . 81
4.17 Closed-loop implementation of the time-varying MPC . . . 83
4.18 Dilation and contraction of the invariant set . . . 84
4.19 Homothetic factor α with respect to the vehicle's position . . . 85
4.20 Closed-loop tracking response . . . 85
4.21 Closed-loop tracking error $\Delta x_2(k)$ . . . 86

4.22 Closed-loop tracking error $\Delta u(k)$ 86

5.1 Example of set of concentric ellipsoids 99
5.2 Smallest ellipsoid $\xi^{(i)}$ that contains the actual state $\Delta x(k)$ 100
5.3 Closed-loop implementation of the robust MPC for LPV systems . 102
5.4 Closed-loop tracking response 102
5.5 Detail of the closed-loop tracking response of Fig. 5.4 103
5.6 Closed-loop tracking error $\Delta x_2(k)$ 103
5.7 Closed-loop tracking error $\Delta u(k)$ 103
5.8 Closed-loop implementation of the explicit approach of the robust MPC for LPV systems 104
5.9 Invariant ellipsoids. 105
5.10 Closed-loop tracking response 106
5.11 Detail of the closed-loop tracking response of Fig. 5.10 106
5.12 Closed-loop tracking error $\Delta x_2(k)$ 107
5.13 Closed-loop tracking error $\Delta u(k)$ 107
5.14 Closed-loop implementation of the MPC for LPV systems using PDLF 118
5.15 Closed-loop tracking response 119
5.16 Detail of the closed-loop tracking response of Fig. 5.22 119
5.17 Closed-loop tracking error $\Delta x_2(k)$ 119
5.18 Closed-loop tracking error $\Delta u(k)$ 120
5.19 Closed-loop implementation of Algorithm 4 120
5.20 Ellipsoids $\xi_1^{(i)}$ 122
5.21 Ellipsoids $\xi_2^{(i)}$ 122
5.22 Closed-loop tracking response 123
5.23 Detail of the closed-loop tracking response of Fig. 5.22 123
5.24 Closed-loop tracking error $\Delta x_2(k)$ 123
5.25 Closed-loop tracking error $\Delta u(k)$ 124
5.26 Performance cost $\gamma(k)$ 125
5.27 Ellipsoids $\xi_1^{(i)}$ 128
5.28 Ellipsoids $\xi_2^{(i)}$ 128
5.29 Closed-loop tracking response in the benchmark 128

5.30 Tracking error $\Delta x_2(k)$ in the benchmark 129
5.31 Control law $\Delta u(k)$ of the explicit algorithm 129
5.32 Input $u(k)$ to the Vir'volt vehicle in the benchmark 129
5.33 Open circuit voltage V_{oc} of the battery 130
5.34 Closed-loop tracking response in the benchmark 131
5.35 Tracking error $\Delta x_2(k)$ in the benchmark 131
5.36 Control law $\Delta u(k)$ of the explicit algorithm 132
5.37 Input $u(k)$ to the Vir'volt vehicle in the benchmark 132

6.1 Periodic driving strategy 149
6.2 Robust adaptive real-time control based on an on-off driving strategy 153
6.3 Deceleration test performed in Saint-Dié-des-Vosges, France, in April 2014 155
6.4 On-board implementation of the robust adaptive real-time control 156
6.5 Real-time response of the embedded robust adaptive control . . . 158
6.6 Detail of Fig. 6.5 158
6.7 Real-time response according to the position 158
6.8 Control response $u(t)$ of the embedded robust adaptive control . . 159
6.9 Open circuit voltage of the battery V_{oc} 159

A.1 Rotterdam's Ahoy circuit 167
A.2 Detail of Rotterdam's Ahoy circuit 168

List of Tables

1.1 Features of the Ahoy circuit 3
1.2 Categories of participation in the Shell Eco-Marathon 4
1.3 Prototype and UrbanConcept vehicle features 5
1.4 Best scores for Electric Battery Prototype 2014 6
1.5 Performances of the EMT in the Shell Eco-Marathon Europe . . . 7
1.6 Off-track awards won by the EMT 8

2.1 Parameters involved in the electric vehicle dynamics 16
2.2 Features of the phases in the Rotterdam's Ahoy circuit 22
2.3 Functions $f_1(\lambda(k))$ and $f_2(\lambda(k))$ 31

4.1 On-line computational resources 78

5.1 On-line computational resources 108
5.2 On-line computational resources 124
5.3 Computational resources of Algorithm 4 embedded in the dsPIC . 127

6.1 Computational resources of Algorithm 8 embedded in the dsPIC . 156
6.2 Comparisons of the official results for the *Vir'volt* prototype at Shell Eco-Marathon of years 2013 and 2014 157

Acronyms

GtL	Gas to Liquid
CNG	Compressed Natural Gas
EMT	EcoMotion Team
ESSTIN	Ecole Supérieure des Sciences et Technologies de l'Ingénieur de Nancy
SoC	State of Charge
PEM	Prediction-Error Minimization
MPC	Model Predictive Control
LPV	Linear Parametric Varying or Linear Parameter-Varying
LMI	Linear Matrix Inequalities
PDLF	Parameter Dependent Lyapunov Function
MIMO	Multiple Input Multiple Output
PMP	Pontryaguin Maximum Principle

ROM Read-only memory

RAM Random-access memory

Notation

n	Order of the system.
p	Number of inputs.
q	Number of outputs.
x_1	Vehicle's position.
x_2	Vehicle's velocity.
x	State-space vector.
u	Control input vector.
I_{batt}	Current of the electric battery.
x_1^*	Optimal reference for the vehicle's position.
x_2^*	Optimal reference for the vehicle's velocity.
x^*	State-space vector of the optimal reference.
I_{batt}^*	Optimal reference for the vehicle's input.
Δx_1	Tracking error of the vehicle's position.
Δx_2	Tracking error of the vehicle's velocity.
Δx	State-space vector of the tracking error.
Δu	Error input.
Δy	Error output.
$\mathbf{A}$	Dynamical matrix.
$\mathbf{B}$	Input Matrix.
$\mathbf{C}$	Output Matrix.
x_{2_e}	Velocity operation point for the linearised system.
I_{batt_e}	Battery current operation point for the linearised system.
m	Vehicle mass.
η	Efficiency of the power inverter.

k_t	Motor constant.
g_r	Transmission gear ratio.
r_w	Radius of the wheels.
ρ	Air density coefficient.
C_d	Aerodynamic drag coefficient.
A_f	Vehicle frontal area.
g	Gravity acceleration coefficient.
C_r	Wheels rolling resistance coefficient.
T_s	Sampling time.
θ	Slope of the road.
$F_{traction}$	Traction force.
$F_{aerodynamics}$	Aerodynamic friction.
$F_{rolling}$	Rolling friction.
F_g	Gravity force.
P_{batt}	Power of the battery.
P_{conv}	Power of the power converter.
P_m	Power of the electric motor.
P_{loss}	Battery losses.
I_m	Current of the electric motor of the vehicle.
T_m	Motor torque.
E_s	Energy stored in the battery.
V_{oc}	Open circuit voltage of the battery.
E_{max}	Maximum energy of the battery.
Q_{max}	Maximum charge capacity of the battery.
$t_{f_{max}}$	Maximum time allowed to finish the race.
t_f	Time spent by the vehicle to finish the race.
$x_{1_{total}}$	Total distance to run in the race.
$I_{batt_{max}}$	Maximum battery current.
$x_{2_{max}}$	Maximum velocity of the vehicle.
F_c	Centrifugal force over the vehicle in curves.
$x_{2_{curve}}$	Velocity of the vehicle in a curve.
$x_{2_{curve_{max}}}$	Maximum safe velocity of the vehicle in a curve.
r_{curve}	Radius of the curve.

F_t	Total wheel-road friction.
P_{mp}	Number of successive phases in the Multi-phase optimal problem.
t_{ini}	Initial time of the phase.
t_{fin}	Final time of the phase.
λ	Parameter of the LPV representation.
λ_{min}	Lower bound of the parameter of the LPV representation.
λ_{max}	Upper bound of the parameter of the LPV representation.
Φ	Polytope of the parameter λ.
$\mathcal{C}$	Polytope of the LPV representation.
$p_{\mathcal{C}}$	Number of vertices in $\mathcal{C}$.
$\mathbf{A}_{1,2}$	Vertices of the polytope $\mathcal{C}$.
$f_{1,2}$	Characteristic functions of the LPV representation.
$\bar{x}$	Steady state target for the state.
$\bar{u}$	Steady state target for the input.
X	Polytopic constraints for the state x.
U	Polytopic constraints for the input u.
X_Δ	Polytopic constraints for the error state Δx.
U_Δ	Polytopic constraints for the error input Δu.
Y_Δ	Polytopic constraints for the error output Δy.
$\mathbf{H}_{X_\Delta}$	Characteristic matrix of the hyperplane representation of the polytope X_Δ.
$\mathbf{H}_{U_\Delta}$	Characteristic matrix of the hyperplane representation of the polytope U_Δ.
$\mathbf{H}_{Y_\Delta}$	Characteristic matrix of the hyperplane representation of the polytope Y_Δ.
$p_{X_\Delta,U_\Delta,Y_\Delta}$	Number of facets in the hyperplane representations of $X_\Delta, U_\Delta, Y_\Delta$, respectively.
J	Performance criterion.
$\mathbf{Q}$	Weighting matrix of the cost of tracking the state.
$\mathbf{R}$	Weighting matrix of the cost of tracking the input.

$\mathbf{P}$	Terminal cost and solution of the Riccati equation.
N_p	Prediction horizon.
X_f	Terminal closed set of feasible final states.
K	Stabilizing state feedback gain derived from the solution of the Riccati equation.
$X_{\Delta u}$	Constraints for the error input Δu in closed loop.
$\mathcal{X}_\Delta$	Polytopic constraints for the error state Δx in closed loop.
$\mathbf{H}_{\mathcal{X}_\Delta}$	Characteristic matrix of the hyperplane representation of the polytope $\mathcal{X}_\Delta$.
$p_\mathcal{X}$	Number of facets in the hyperplane representation of $\mathcal{X}_\Delta$.
Ω	Invariant set.
$\mathbf{V}_\Omega$	Matrix of vertices of the vertex representation of the polytope Ω.
$\mathbf{w}$	Characteristic matrix of the vertex representation of the polytope Ω.
p_Ω	Number of vertices in the vertex representation of Ω.
$\mathcal{X}_{\Delta_{-k}}$	Pre-image of $\mathcal{X}_\Delta$.
$\hat{X}_\Delta$	Nominal polytopic constraints for the error state Δx.
$\hat{U}_\Delta$	Nominal polytopic constraints for the error input Δu.
$\mathbf{H}_{\hat{X}_\Delta}$	Characteristic matrix of the hyperplane representation of the polytope $\hat{X}_\Delta$.
$\mathbf{H}_{\hat{U}_\Delta}$	Characteristic matrix of the hyperplane representation of the polytope $\hat{U}_\Delta$.
$\hat{\Omega}$	Nominal invariant set.
$\mathbf{V}_{\hat{\Omega}}$	Matrix of vertices of the vertex representation of the polytope $\hat{\Omega}$.
$p_{\hat{\Omega}}$	Number of vertices in the vertex representation of $\hat{\Omega}$.
v_j	*j-th* vertex of the polytope $\hat{\Omega}$.
$\mathcal{P}_i$	*i-th* hyperplane of the polytope $\mathcal{X}_\Delta$.
H_i	*i-th* characteristic matrix of the hyperplane representation of the polytope $\mathcal{X}_\Delta$.

β	Homothetic multiplier.
α	Homothetic factor.
Δu_{max}	Symmetric constraint for Δu.
Δy_{max}	Symmetric constraint for Δy.
F	State feedback gain.
V	Lyapunov function.
γ	Upper bound of the Lyapunov function V.
$\mathbf{\Theta}, \mathbf{\Lambda}$	Solutions of a convex optimization problem involving LMIs.
$\mathbf{M}$	Weighting gain of the Lyapunov function.
ξ	Asymptotically stable invariant ellipsoidal set.
$\Delta\check{x}$	Set of gridded state-space.
$\check{\mathbf{\Theta}}$	Set of solutions of a convex optimization problem involving LMIs.
$\check{\mathbf{\Lambda}}$	Set of solutions of a convex optimization problem involving LMIs.
$\check{\xi}$	Set of invariant ellipsoids.
$\mathbf{\Theta}_{1,2}$	Solution of a convex optimization problem involving LMIs.
$\mathbf{\Lambda}_{1,2}$	Solutions of a convex optimization problem involving LMIs.
$\mathbf{G}_{1,2}$	Solution of a convex optimization problem involving LMIs.
$\xi_{1,2}$	Ellipsoidal set.
Ω_1	Asymptotically stable invariant set.
a, b, c	Coefficients of a first order differential equation.
$\overrightarrow{\mathcal{V}_w}$	Vector of the velocity of the wind.
$\overrightarrow{x_2}$	Vector of the velocity of the vehicle.
$\overrightarrow{v}$	Component of $\overrightarrow{\mathcal{V}_w}$ that is collinear with $\overrightarrow{x_2}$.
$a_{(1)}, b_{(1)}, c_{(1)}$	Initial solution of a first order differential equation.
s^*	Affine multiplier.
$a_{(s^*)}, b_{(s^*)}, c_{(s^*)}$	Affine solution of a first order differential equation.
$C(u(t))$	Energetic cost of the control $u(t)$.

C_{on}	Energetic cost of switching on the electric motor of the vehicle.
V_{min}	Minimum value of the velocity with a periodic on-off control.
V_{max}	Maximum value of the velocity with a periodic on-off control.
$\bar{V}$	Average velocity with a periodic on-off control.
$\bar{V}^*$	Target average velocity.
t_{on}	Time during which the motor is on.
t_{off}	Time during which the motor is off.
$\check{V}_{min}$	Set of minimum velocities for an on-off control.
$\bar{1}$	Ones vector.
$\mathbf{I}$	Identity matrix.
$\mathbb{N}$	Set of natural numbers.
$\mathbb{R}$	Set of real numbers.

Chapter 1

General Introduction

In the field of transportation, the research on energy efficiency has been carried out for few decades by the automotive industry, where one of the main requirements is to reduce harmful emissions. A solution to tackle this problem is the implementation of an alternative energy source (electric, solar, hydrogen, fuel, etc.). The main issue is the problem of how the energy source must be used in order to maximize the energy efficiency [61, 18]. The main objective is to reduce the energetic consumption.

Achieving a low consumption requires the solution of three central tasks: the modelling of the problem, the computation of a low consumption strategy, and finally the real-time implementation [61]. First, a suitable model of the vehicle must be obtained. Secondly, and of special interest, is the problem of how using different energy sources, one or several, so that the energy efficiency can be maximized. This particular problem can be rephrased as how the vehicle must be driven so that the minimum quantity of energy is used, this is the *optimal driving strategy* [61, 60, 18]. The reference driving trajectory must be derived in terms of expected position and velocity all along the path to run. The first and second tasks can be performed off-line. In the third and final task, a powerful tracking strategy must be designed to guarantee that the driving strategy is accomplished. The tracking strategy must cope with the problem constraints, taking into account the inherent limitations of the real-time implementation, such as computing time and memory resources [61, 50].

The process described above, from the modelling and description of the prob-

lem to the real-time implementation, can be carried out for a particular task, and tested rigorously within academic frameworks such as those provided by sustainable vehicle competitions all around the world. This competitions such as The Shell Eco-Marathon [3], The Zero race [7] (now The WAVE Trophy [6]), The EcoCar 2 [1], among others, invite industries, universities and research groups to innovate, in a regular basis, in solutions to the problem of sustainable transportation and the efficient use of the different energy sources.

1.1 The European Shell Eco-Marathon

The European Shell Eco-Marathon is a race involving ecological and economical vehicles. It brings together nearly 200 teams from high schools and universities coming from all over Europe (See Fig. 1.1). The aim of the contest is to promote the research and innovation in the field of sustainable and environmental transportation. The principle of the race is to drive a fixed number of kilometres in a limited range of time with the least possible consumption of fuel.

Figure 1.1: Shell Eco-Marathon Europe 2014. This year participated 198 teams from 27 countries. Picture taken from the Shell Eco-Marathon 2014 website in Flickr [5].

The Shell Eco-Marathon has its origins back in 1939 in a Shell research laboratory in Illinois, USA. The race started as a competition between partner scientists

trying to drive the longer distance with the least quantity of fuel. Back then the record was 21 kilometres driven with only one litre of fuel. Several years later, the European Shell Eco-Marathon, as we know it today, begun in France in 1985 with a record of 680km/l [61].

Since the year 2012, the European Shell Eco-Marathon race has been held in Rotterdam, The Netherlands, in the Ahoy circuit (see Fig. A.1). The main features of the Ahoy circuit are summarized in Table 1.1. For further details on the dimensions and features of the Ahoy circuit, please refer to Appendix A.

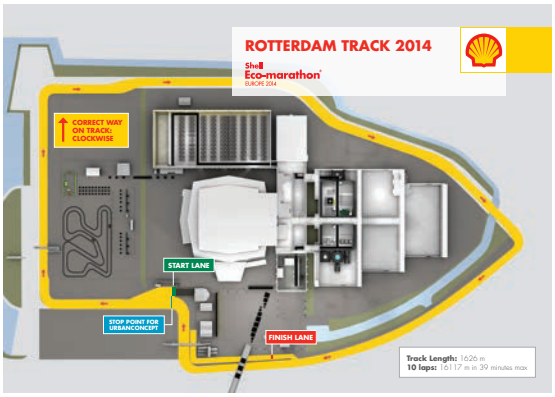


Figure 1.2: Rotterdam's Ahoy circuit 2014 [3].

Table 1.1: Features of the Ahoy circuit.

Lap length	1.626km
Number of laps to make a run	10
Total distance	16.26km
Time allowed to the run	39min
Number of 90° turns	5

1.1.1 Categories of participation

There are six different categories of participation in the race, which are distinguished according to the source of energy used, as indicated in Table 1.2. The teams are allowed to use only one of the energy sources indicated in Table 1.2 according to the Shell Eco-Marathon rules available in [4]. Although each category

Table 1.2: Categories of participation in the Shell Eco-Marathon.

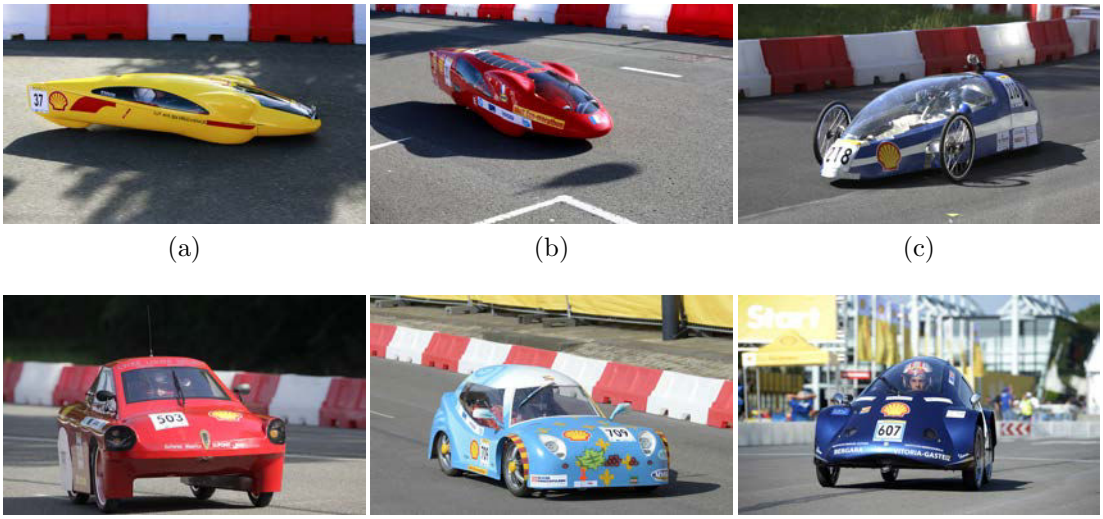
Energy source		**Measurement of the consumption**
Internal Combustion	Gasoline	kilometre per litre [km/l]
	Diesel	kilometres per litre [km/l]
	Ethanol E100	kilometres per cubic meter of ethanol [km/m^3]
	Shell GtL - Gas to Liquid	kilometres per cubic meter of ethanol [km/m^3]
	CNG - Compressed natural gas (Pure methane)	kilometres per cubic meter of methane [km/m^3]
Electric Mobility	Battery Electric	kilometres per kilowatt hour [km/kWh]
	Hydrogen	kilometres per cubic meter of hydrogen [km/m^3]

has its own expression of energy consumption, the final result is converter into an equivalent measure in terms of kilometres per litre of fuel or diesel for comparison purposes. To compare, for example, the consumption in kilometres per kilowatt hour with the consumption in kilometres per litre of fuel (Shell FuelSave Unleaded 95) the following formula can be used

$$1\text{km/kWh} = 8.892\text{km/l}, \tag{1.1}$$

assuming that the density of the fuel is 0.74616kg/l at 15°C, and therefore the energy of one litre of fuel is 32010kJ which is equivalent to 8.892kWh [4].

The vehicles must be fully designed and built by the teams themselves. For each class of energy source, the vehicle can be a Prototype style vehicle or UrbanConcept style vehicle. The first ones are small vehicles of three or four wheels and usually the driver is laying down (see Fig. 1.4a - 1.4c). The second ones are bigger vehicles of four wheels and look like small commercial type passenger cars (see Fig. 1.3d - 1.3f). Both of them are single-seater cars. In Table 1.3, the main features of the two classes of vehicle are presented.



(a) (b) (c)

(d) (e) (f)

Figure 1.3: Prototype and UrbanConcept vehicles. (a) Prototype Gasoline. (b) Prototype Battery Electric. (c) Prototype Hydrogen. (d) UrbanConcept Gasoline. (e) UrbanConcept Battery Electric. (f) UrbanConcept Hydrogen. Pictures taken from the Shell Eco-Marathon 2014 website in Flickr [5].

Table 1.3: Prototype and UrbanConcept vehicle features [4].

Feature	**Prototype**	**UrbanConcept**
Number of wheels	3 or 4	4
Vehicle height	$\leq$ 100cm	$\geq$ 100cm and $\leq$ 130cm
Vehicle length	$\leq$ 350cm	$\geq$ 220cm and $\leq$ 350cm
Vehicle width	$\leq$ 130cm	$\geq$ 120cm and $\leq$ 130cm
Number of pilots	1	1
Weight of the pilot	$\geq$ 50kg	$\geq$ 70kg
Weight of the vehicle (without the pilot)	$\leq$ 140kg	$\leq$ 225kg
Space for luggage required	No	Yes
Luggage dimension (L×H×W)	-	500 × 400 × 200mm

1.1.2 The Shell Eco-Marathon around the world

The Shell Eco-Marathon also exists in America and Asia. The Shell Eco-Marathon Americas was launched in 2007 in the United States, involving teams from Canada to Brazil. The Shell Eco-Marathon Asia started in 2010, in Malaysia [3]. In Table 1.4, the Shell Eco-Marathon 2014 results for Electric Battery Prototype

are presented.

Table 1.4: Best scores for Electric Battery Prototype 2014.

Event	Number of paticipants in the category	Best score	Team	Country
Europe	28	1091.6km/kWh	Graz University of Technology	Austria
Asia	14	263.4km/kWh	Rattanakosin Technological College	Thailand
Americas	10	537.2m/kWh	Mater Dei High School	United States

1.2 The EcoMotionTeam

The EcoMotion Team (EMT) of the Ecole Supérieure des Sciences et Technologies de l'Ingénieur de Nancy (ESSTIN) in France, has been involved in the European Shell Eco-Marathon race from the year 2000 with prototypes in the categories gasoline, hydrogen and battery electric. In Fig. 1.4 are presented several of the prototypes developed by the EMT. The performances of the EMT are presented since the year 2000 in Table 1.5 .

(a) (b) (c)

Figure 1.4: Prototypes developed by the EMT. (a) *Mirabelle* Gasoline Prototype, 2004. (b) *HydrogESSTINe* Hydrogen Prototype, 2008. (c) *Vir'Volt* Battery Electric Prototype, 2014.

Table 1.5: Performances of the EMT in the Shell Eco-Marathon Europe since the year 2000 [2] (*no ranked result).

Year	**City**	**Category**	**Vehicle**	**Performance**	**Rank**
2000	Castellet (FRA)	Gasoline Prototype	*ESSTIN1*	330km/l	80^{th}/150
2001	Nogaro (FRA)	Gasoline Prototype	*Ern'ESSTIN*	288km/l	99^{th}/150
2002	Nogaro (FRA)	Gasoline Prototype	*Destinée*	182km/l	82^{th}
2003	Nogaro (FRA)	Gasoline Prototype	*Mirabelle*	812km/l	30^{th}
		Hydrogen Prototype	*Combustine*	*	*
2004	Nogaro (FRA)	Gasoline Prototype	*Mirabelle*	810km/l	30^{th}
		Hydrogen Prototype	*Combustine*	1929km/l	2^{nd}
2005	Nogaro (FRA)	Hydrogen Prototype	*HydrogESSTINe*	1894km/l	3^{rd}/9
2006	Nogaro (FRA)	Hydrogen Prototype	*HydrogESSTINe*	2784km/l	1^{st}/9
		Solar Prototype	*HélioSSTINe*	191626J	2^{nd}/6
2007	Nogaro (FRA)	Hydrogen Prototype	*HydrogESSTINe*	*	*
		Solar Prototype	*HélioSSTINe*	*	*
2008	Nogaro (FRA)	Hydrogen Prototype	*HydrogESSTINe*	2509, 12km/l	4^{th}
2009	Lausitz (DEU)	Hydrogen Prototype	*HydrogESSTINe*	2814km/l	4^{th}

2010	Lausitz (DEU)	Hydrogen Prototype	*HydrogESSTINe*	2379km/l	5^{th}
2011	Lausitz (DEU)	Battery Electric Prototype	*Vir'Volt*	532km/kWh	2^{nd}/12
2012	Rotterdam (NLD)	Battery Electric Prototype	*Vir'Volt*	440.5km/kWh	8^{th}/21
2013	Rotterdam (NLD)	Battery Electric Prototype	*Vir'Volt*	638.5km/kWh	6^{th}/27

Additionally to the ranking obtained in the different categories of energy sources, the EMT has also won several off-track awards. The off-track awards are given in areas such as the technical innovation, the safety, the communication and the pedagogic process held by the students in the team (National education award). In Table 1.6 are presented the off-track awards won by the EMT since its beginnings in the year 2000.

Table 1.6: Off-track awards won by the EMT since the year 2000 [2].

Year	**Off-track award**
2003	1^{st} place of the National education award
2004	2^{nd} place of the Technical Innovation award
2005	2^{nd} place of the Communication award
	3^{th} place of the Safety award
2006	2^{nd} place of the Technical Innovation award
	2^{nd} place of the National education award
	3^{rd} place of the Communication award
2008	1^{st} place of the Technical Innovation award
	1^{st} place of the Safety award
2009	1^{st} place of the Technical Innovation award
2010	3^{th} place of the Safety award
2013	2^{nd} place of the Safety award
2014	1^{st} place of the Safety award

1.3 Motivation of the work

Initially, the Driving strategy proposed by the EcoMotionTeam was so far a stop-and-go one, that merely reduced to manually turn on or off the engine by the pilot according to whether the vehicle was going up or down, or to drive around an average value, the efficiency of the converter/motor being better when working at full regime. It was typically the strategy chosen in 2011 when the race was held in Lausitz, Germany. Indeed, the Lausitz's track could be considered as a straight line since there was no deceleration or complex curves. Hence, the driving instructions corresponding to this simple strategy could be easily followed by the pilot. The new and ambitious challenges in terms of consumption require more complex driving strategies and the accuracy of the tracking is decisive in the final performances. The main objective of this work is to propose robust low consumption driving strategies for the *Vir'volt* prototype. The main issue is the capability of those controlled driving strategies to be embedded into the digital devices available in the vehicle. The synthesis of the embedded controllers is subject to constraints such as the limited memory and computational capacities of the electronic equipment.

1.4 Outline

The present work is organized as follows.

- In Chapter 2, a model of the vehicle dynamics is obtained. The unknown parameters involved in the vehicle dynamics are estimated using parameter identification from experimental data. The identified dynamics is used to derive an optimal driving strategy that is intended to be tracked on-line during the driving task. The tracking problem is formulated using either a linearised model around an operating point or a Linear Parametric Varying (LPV) representation. A benchmark, designed during the thesis and used to test the performances of the tracking strategies is described.

- In Chapter 3, background on the problem of the MPC-based tracking strat-

egy for LTI discrete-time systems is recalled. The tracking task is subject to time-invariant polytopic constraints on the input and/or the state. The asymptotic stability of the control law is guaranteed by resorting to invariant set theory. The performances of the MPC tracking strategy are assessed with first, a numerical example and then, with the *Vir'volt* vehicle.

- In Chapter 4, the problem of MPC tracking problem subject to time-varying polytopic constraints is presented. The asymptotic stability of the control law is guaranteed by a homothetic time-varying invariant set. The real-time capacity of the resulting MPC tracking strategy is assessed again with a numerical example and then with the *Vir'volt* vehicle.

- In Chapter 5, the problem of the MPC tracking for a Linear Parametric Varying (LPV) representation of the vehicle is proposed. The LPV description allows to capture the nonlinearities of the dynamics. In this chapter, a new explicit MPC strategy for LPV systems is developed. The stability is guaranteed by the use of a Parameter Dependent Lyapunov Function (PDLF) to take into account the time-varying parameter to reduce the conservatism. The contribution here is to propose an approach well suitable for real-time applications, since this approach does not involve the on-line solution of any Linear Matrix Inequality (LMI) in the computation of the control law. The performances of the approach are tested on the benchmark for the *Vir'volt* vehicle.

- In Chapter 6, a robust real-time adaptive control is developed for the vehicle. This approach is presented as a complementary scheme of driving strategy, in which the dynamics of the vehicle is identified on-line and the derived driving strategy is an on-off strategy. The on-line identification of the dynamics guarantees that the disturbances are taken into account. The on-off driving strategy is computed on-line according to the identified dynamics, and aims to reduce the energetic consumption required to perform the task. The robust adaptive control has been embedded in a dsPIC device and has been tested during the European Shell Eco-Marathon 2014.

- Finally, general conclusions and perspectives are presented in Chapter 7.

Chapter 2

The Low Consumption Vir'volt Electric Vehicle

2.1 Introduction

The EcoMotionTeam has developed successive vehicles over the past 15 years. The electric prototype considered in this work is named *Vir'Volt* and is the fifth generation. This prototype, shown in Fig. 2.1, has been ranked 2^{nd}, in 2011, in the Battery Electric category among 12 others vehicles and 7^{th} among the 100 participants of the European competition with a result of 532km/kWh (equivalent to 4731km with one litre of fuel, calculated using (1.1)). The prototype is a three

Figure 2.1: The *Vir'Volt* prototype in the Shell Eco-Marathon 2011.

wheels vehicle. The direction is controlled by the front wheel. In the years 2011 and 2012, the propulsion was given by one of the two rear wheels. In the years

2013 and 2014, the propulsion is given by the front wheel. It can reach the speed of 35km/h. The energy is provided by a 24V battery to an electric motor which develops $0,4$Nm. The vehicle has many embedded electronics devices in order to communicate with the pit stop. The total weight of the car is 40kg, the pilot needs to weight at least 50kg according to the Shell Eco-Marathon rules. Thus, the total mass of the vehicle is 90kg.

In this chapter, the full procedure performed to achieve a low consumption strategy for the *Vir'volt* electric vehicle, intended to participate in the European Shell Eco-Marathon, is presented. In Section 2.2, the modelling of the vehicle is described. Parameter identification is used in order to estimate the unknown parameters involved in the vehicle dynamics. In Section 2.3, the low consumption strategy is detailed. This strategy indicates the vehicle velocity and the battery current required to achieve the minimum energetic consumption with respect to the position of the vehicle in the circuit. The driving strategy is the result of an optimization problem and it is intended to be tracked during the driving task. In Section 2.4, a discrete-time model of the vehicle dynamics is obtained in order to allow the design of a discrete-time control law that will be embedded on-board in a digital device. In Section 2.5, the tracking problem is formulated. The nonlinear tracking error is derived using first, a linearised model around an operating point and, next, a Linear Parametric Representation (LPV). Finally, is Section 2.6, the benchmark which will be used for testing the performances of the different control strategies is described. The benchmark allows to emulate the vehicle and the circuit profile during the race.

2.2 The Vir'volt prototype

2.2.1 Electric vehicle dynamics

The dynamics of the vehicle can be described in terms of the force of traction $F_{traction}$[N], the external forces due to the aerodynamics $F_{aerodynamics}$[N], the rolling (contact wheel-ground) resistance $F_{rolling}$[N], and the slope resistance F_{slope}[N] due to the vehicle's weight and the road slope θ[rad] (see Fig. 2.2) [60, 32]. All this forces are related by Newton's laws of motion involving the mass m[kg]

of the vehicle and its acceleration $\frac{dx_2(t)}{dt}[\mathrm{m/s^2}]$ as follows

$$m\frac{dx_2(t)}{dt} = F_{traction}(t) - F_{aerodynamics}(t) - F_{rolling}(t) - F_g(t), \tag{2.1}$$

with $x_2(t)[\mathrm{m/s}]$ the velocity of the vehicle.

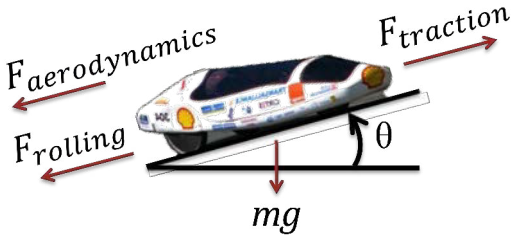


Figure 2.2: Free-body diagram.

The aerodynamic force $F_{aerodynamics}$ is given in terms of the frontal area $A_f[\mathrm{m^2}]$ of the vehicle, the aerodynamic drag coefficient C_d and the air density $\rho[\mathrm{kg/m^3}]$. Thus,

$$F_{aerodynamics}(t) = \frac{1}{2}\rho C_d A_f x_2(t)^2. \tag{2.2}$$

The rolling force $F_{rolling}$, for a road with variable slope $\theta[\mathrm{rad}]$ (assumed positive if the vehicle is going uphill and negative if it is going downhill), is given in terms of the wheels rolling resistance coefficient C_r and gravitation acceleration $g[\mathrm{m/s^2}]$, as follows

$$F_{rolling}(t) = mgC_r\cos(\theta(t)). \tag{2.3}$$

The gravitational force F_g due to the slope of the road and to the weight of the vehicle reads

$$F_g(t) = mg\sin(\theta(t)). \tag{2.4}$$

Finally, in the *Vir'volt* electric vehicle, the power-train configuration (see Fig. 2.3) is composed by a Kypom 22.2V battery that feeds a DC Maxon 200Watt motor which develops $0,4$Nm. The torque of the motor is transmitted to one of the rear wheels by a torque coupler. Therefore, the traction force $F_{traction}$ is given by

$$F_{traction}(t) = \frac{\eta k_t g_r}{r_w} I_{batt}(t), \tag{2.5}$$

with $\eta = I_m/I_{batt}$ the efficiency of the power converter, $k_t = T_m/I_m$[Nm/A] the motor constant given by the motor manufacturer (I_m[A] is the motor current and T_m[Nm] the motor torque), I_{batt}[A] the battery current, g_r the transmission gear ratio and r_w[m] the radius of the wheel.

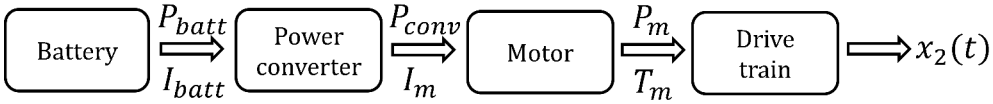


Figure 2.3: *Vir'volt* Power train. I_{batt}[A] and I_m[A] are the battery and motor currents, respectively. T_m[Nm] the motor torque. P_{batt}[W], P_{conv}[W] and P_m[W] are the power of the battery, the power of the converter and the motor power, respectively. x_2[m/s^2] is the vehicle velocity.

By plugging (2.2), (2.3), (2.4) and (2.5) into (2.1), the following dynamics for the vehicle is obtained:

$$m\frac{dx_2(t)}{dt} = \frac{\eta k_t g_r}{r_w} I_{batt}(t) - \frac{1}{2}\rho C_d A_f x_2(t)^2 - mgC_r\cos(\theta(t)) - mg\sin(\theta(t)). \quad (2.6)$$

Since the battery provides all the traction power, the battery current I_{batt} is the input of the system. The internal frictions have been neglected. The rotational inertias in the power-train, such as the rotor inertia, have been neglected by being smaller than the vehicle mass.

The distance x_1[m] travelled by the vehicle, considered hereafter as the actual position of the vehicle, reads

$$\frac{dx_1(t)}{dt} = x_2(t). \quad (2.7)$$

2.2.2 Parameter identification

In the actual real-time application, the dynamics (2.6) is partially unknown since the values of the parameters η, C_r and the product $C_d A_f$ are unknown. However, the model can be identified from experimental data by performing a *nonlinear grey-box system identification* [34] (see Fig. 2.4).

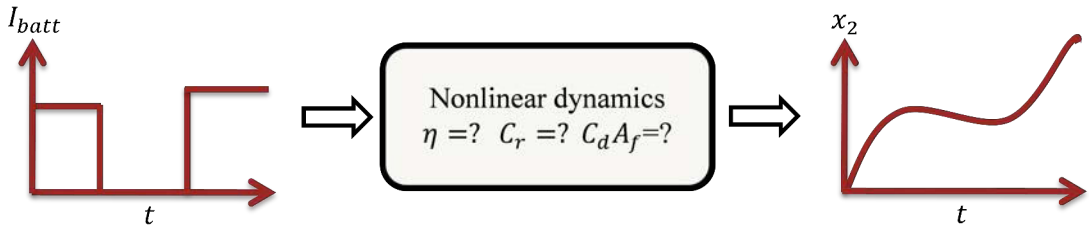


Figure 2.4: Grey-box nonlinear identification.

2.2.2.1 Nonlinear grey-box identification

In System Identification, the mathematical model of a system can be estimated by using three kinds of information: the knowledge on the structure of the model (structural knowledge) or the physical knowledge obtained from first principles, the data taken from intentional experiments performed on the system, and the assumptions made over the validity of the model [34].

If the model of the system is found using only structural knowledge or physical knowledge, then the model is a *white-box* model [34]. The *white-box* models are usually described by differential equations. If the models are estimated to fit the experimental data regardless of the structure of the model, then they are *black-box* models [34]. In between are the *grey-box* models, in which the structure of the model is known a priori and the data are used to estimate the values of the unknown parameters of the model [34, 48].

The System Identification Toolbox™ of Matlab®allows to perform grey-box identification. The continuous-time nonlinear differential equations of the model having been properly defined, the unknown parameters can be identified using iterative Prediction-Error Minimization techniques (PEM) for continuous-time linear and nonlinear models [48].

2.2.2.2 Parameter estimation

To perform the estimation of the unknown parameters η, C_r and C_dA_f, several accelerating and decelerating tests were performed to collect the data required for the identification process. In Fig. 2.5, the initial experimental data for a flat path ($\theta = 0$) used to perform the nonlinear grey-box identification are depicted

in black.

The known parameters are $k_t = 0.0604\text{Nm/A}$, $g_r = 8.5$, $m = 90\text{kg}$, $r_w = 0.24\text{m}$, $\rho = 1.225\text{kg/m}^3$ and $g = 9.81\text{m/s}^2$, and the known continuous-time non-linear structure of the model is the one given by (2.6). Then, the parameters η, C_r and the product C_dA_f have been identified with the PEM method. The best fit to the experimental data has given

$$\eta = 0.97,\ C_dA_f = 0.1031\text{m}^2 \text{ and } \text{C}_\text{r} = 8.1549 \times 10^{-4}. \tag{2.8}$$

The parameters involved in the dynamics (2.6) are summed up in Table 2.1.

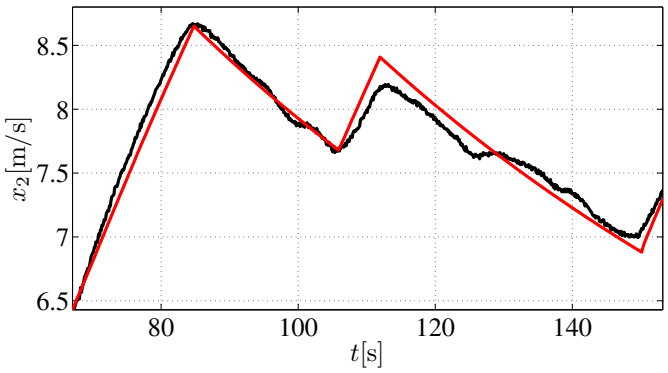
Table 2.1: Parameters involved in the electric vehicle dynamics (2.6).

Coeff.	**Description**	**Value**
m	vehicle mass	90kg
η	efficiency of the inverter	0.97
k_t	motor constant	0.0604Nm/A
g_r	transmission gear ratio	8.5
r_w	radius of the wheels	0.24m
ρ	air density coefficient	1.225kg/m^3
C_dA_f	aerodynamic drag coefficient × vehicle frontal area	0.1031m^2
g	gravity acceleration coefficient	9.81m/s^2
C_r	rolling resistance coefficient	8.1549×10^{-4}

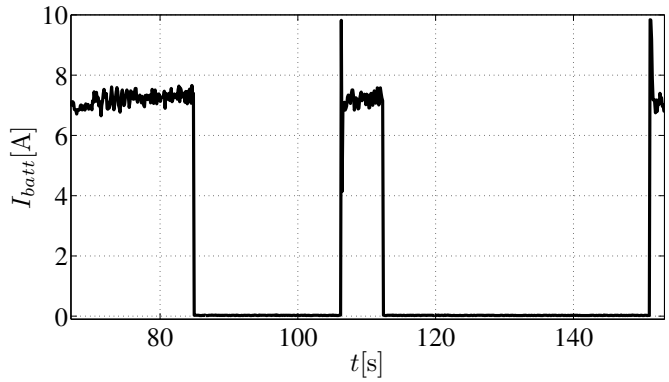
2.3 Low consumption driving strategy

The dynamics of the vehicle being fully identified, the problem of the energy-management can now be addressed. The problem amounts to defining how using the available energy sources so that the energy efficiency can be maximized. The previous question can be rephrased as how the vehicle must be driven so that the minimum quantity of energy is used during the driving task (a driving task can be single or multiple repetitions of a prescribed circuit, or a common route [61]). The answer to this questions is precisely the *driving strategy* [60].

Since the battery provides all the traction power (see Fig. 2.3), the problem of the low consumption strategy becomes an electrical resource management prob-



(a)



(b)

Figure 2.5: Grey-box identification of the nonlinear dynamics (2.6) for a flat path. (a) Velocity data. In black is the data used for the identification and in red is the velocity response of the identified model. (b) Input data.

lem, where the energy level of the battery (only the discharging of battery being taken into account) is the critical variable in the formulation of the optimization problem that leads to the driving strategy solution [41, 61].

2.3.1 Energetic considerations

In a general way, the energy stored in the battery at time t can be expressed as follows

$$E_s(t) = E_s(0) - \int_0^t P_{batt}(\tau) - P_{loss}\, d\tau, \tag{2.9}$$

where $E_s(0)$ is the initial stored energy, P_{batt} is the power delivered to the load by the battery and P_{loss} are the battery losses [41]. Disregarding the losses, the battery energy can be simply expressed in terms of the current flowing through the battery I_{batt} and the constant open circuit voltage V_{oc} as [27, 41]

$$E_s(t) = E_s(0) - \int_0^t V_{oc} I_{batt}(\tau)\, d\tau. \tag{2.10}$$

The dynamics of the energy in the battery can also be expressed in term of the State of Charge of the battery. In [41], the State of Charge (SoC) of the battery at time t is defined as the following relative energy level (scalar)

$$SoC(t) = \frac{E_s(t)}{E_{max}}, \tag{2.11}$$

where E_{max} is the energy capacity of the battery fully charged. The energy stored in the battery at time t can also be expressed in terms of the battery voltage V_{oc} and the battery charge $Q(t)$ (given by the ampere-hour battery capacity), i.e.

$$E_s(t) = V_{oc} Q(t) \text{ and } E_{max} = V_{oc} Q_{max} \tag{2.12}$$

where Q_{max} is the maximum charge capacity of the battery. By plugging (2.11) and (2.12) into (2.10), the SoC remaining in the battery at time t from an initial SoC value (SoC_0) is expressed by [57, 27]

$$SoC(t) = SoC_0 - \frac{1}{Q_{max}} \int_0^t I_{batt}(\tau)\, d\tau. \tag{2.13}$$

The method presented in (2.13) to estimate the SoC value is known as *Ampere-hour-counting* (Ah), and is very suitable for electric vehicle applications due to the regular full charges before the driving cycles. In fact, as [57] highlights, to avoid dealing with the battery ageing, the full SoC is considered when the battery charge has not changed during the last two hours at constant voltage and temperature.

From the right side of (2.10) and (2.13), it is clear that the lowest consumption, or equivalently the highest energy left in the battery at the end of the task, is

achieved if the minimum electric charge, i.e. the minimum integral $I^*_{batt}(t)$ of battery current, is used to perform the driving task.

2.3.2 Optimization problem

In this section, we consider the off-line computation of the reference driving strategy which must achieve the minimization of the energy consumption taking into account some constraints due to the dynamics and to the race path.

For a prescribed circuit, considered as a succession of curves and straight lines (see Fig. 2.6), the driving strategy is a finite collection of triplets $(x^*_1(t), x^*_2(t), I^*_{batt}(t))$, where $x^*_2(t)$ corresponds to the required velocity assigned to the position $x^*_1(t)$ in the circuit at time t, such that the minimal consumption is achieved.

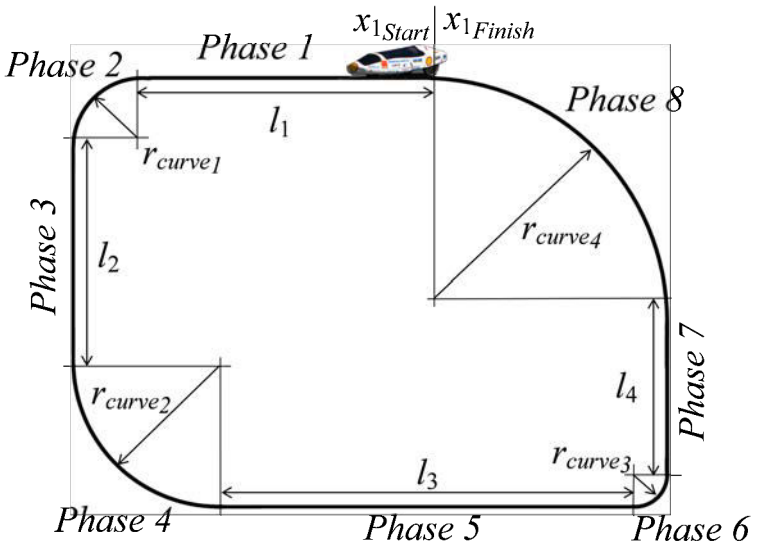


Figure 2.6: Example of a track with straight lines and curves. l_i[m] is the length of the *i-th* straight line and r_{curve_j}[m] is the radius of the *j-th* curve.

2.3.2.1 Constraints of the Optimality problem

The search for the driving strategy is an optimization problem which, given the model of the vehicle, the road profile (slope, curves, straight lines, etc.) and the constraints in terms of maximum velocity allowed at each curve, maximum time

of the race, and total number of kilometres, must provide $(x_1^*(t), x_2^*(t), I_{batt}^*(t))$ at time t. The optimization problem must consider the following constraints:

- The maximum time $t_{f_{max}}$[s] allowed to complete the race.
- The total distance $x_{1_{total}}$[m] to run.
- The maximum battery current $I_{batt_{max}}$[A].
- The maximum speed $x_{2_{max}}$[m/s^2] for the vehicle.
- The limits of the centrifugal force F_c[N].

In a curve, the centrifugal force $F_c(N)$ over the car is $F_c = m(x_{2_{curve}})^2/r_{curve}$, with r_{curve}[m] the radius of the curve (see Fig. 2.6) and $x_{2_{curve}}$[m/s^2] the velocity of the vehicle in the curve. This force must not be larger than the total frictions forces wheel-road F_t[N] to prevent the car slipping over, i.e. $x_{2_{curve}} \leq \sqrt{F_t r_{curve}/m}$ [60].

Notice that the constraint over the velocity $x_{2_{curve}}$ in the curves is concerned exclusively to the position x_1 where there is precisely a curve, otherwise the constraint is only $x_2 \leq x_{2max}$. As a result, one has

$$\begin{aligned} &x_2(t) \leq x_{2max}, \text{ if the vehicle is in a straight line,} \\ &x_2(t) \leq \sqrt{F_t r_c/m} \leq x_{2max}, \text{ if the vehicle is in a curve.} \end{aligned} \tag{2.14}$$

The optimality problem that includes the aforementioned constraints and that leads to the driving strategy for driving through a known path in a finite time t_f, with the minimum electrical consumption, is stated in the following.

Optimal control problem 1

$$\begin{aligned} \min_{I^*_{batt}, x^*_1, x^*_2, t_f} \ & \int_0^{t_f} I_{batt}(t)\, dt \\ s.t. \quad & (2.6), \\ & (2.7), \\ & x_1(0) = 0, \\ & x_2(0) = 0, \\ & t_f \leq t_{f_{max}}, \\ & x_1(t_f) = x_{1_{total}}, \\ & 0 \leq x_2(t) \leq x_{2_{max}}, \forall t \leq t_f, \\ & (2.14), \\ & I_{batt}(t) \in [0, I_{batt_{max}}], \forall t \leq t_f. \end{aligned} \tag{2.15}$$

The optimal values of the control input $I^*_{batt}(t)$ and the optimal trajectory $x^*_2(t)$ are solution of the problem (2.15) for a given position $x^*_1(t)$. To solve (2.15) involving the constraints (2.14), the optimality problem (2.15) is represented as a multi-phase optimality problem, as detailed in next subsection.

2.3.2.2 The Multi-phase Optimality problem

To include appropriately the constraints (2.14) in the solution of the optimality problem (2.15), it is divided in consecutive phases of straight lines and curves, according to the circuit shape. For each phase, the problem (2.15) is solved with the corresponding constraint (2.14), depending on whether the vehicle is in a curve or in a straight line.

Consider for example the road in Fig. 2.6, with four straight lines and four curves. Then, eight consecutive phases are obtained for one lap. The optimality problem (2.15) is expressed as a multi-phase optimality problem of size $P_{mp} = 8$ i.e.

$$\min \int_0^{t_f} I_{batt}(t)\, dt = \min \sum_{j=1}^{P_{mp}} \int_{t^{(j)}_{ini}}^{t^{(j)}_{fin}} I_{batt}(t)\, dt, \tag{2.16}$$

with P_{mp} the number of consecutive phases, $t^{(j)}_{ini}$ the time in which the *j-th* phase starts, and $t^{(j)}_{fin}$ the time in which the *j-th* phase finishes, with $\sum_{n=1}^{P_{mp}} (t^{(j)}_{fin} - t^{(j)}_{fin}) =$

t_f [36, 58]. The constraints (2.14) are rewritten as

$$\begin{aligned} x_2(t) &\leq x_{2max} \text{ if the } j\text{-}th \text{ phase is a straight line,} \\ x_2(t) &\leq \sqrt{F_t r_c/m} \text{ if the } j\text{-}th \text{ phase is a curve.} \end{aligned} \tag{2.17}$$

The GPOPS Toolbox™ of Matlab®, allows to solve Multi-phase optimal control problems for linear and nonlinear dynamics, in continuous-time or discrete-time [36, 58]. For this purpose, the following must be properly defined for each phase of the track

- The dynamics of the system given by (2.6) and (2.7).
- The cost functional as in (2.16).
- The dimensions of each phase (the length of the straight lines and the radius and angle of each curve).
- The optimization constraints.

The present work focuses in the Rotterdam's Ahoy circuit, where the race took place in the years 2012 and 2013. In this circuit five curves of 90° are connected by five straight lines (or portions considered as straight lines). Therefore the circuit is divided in ten consecutive phases (See Fig. 2.7). The features of each phase are presented in Table 2.2.

Table 2.2: Features of the phases in the Rotterdam's Ahoy circuit.

Phase	**Description**	**Dimensions**
Phase 1	Straight Line	Length 435m
Phase 2	90° curve	Radius 25m
Phase 3	Straight Line	Length 639m
Phase 4	90° curve	Radius 21.3m
Phase 5	Straight Line	Length 250m
Phase 6	90° curve	Radius 16.3m
Phase 7	Straight Line	Length 148m
Phase 8	90° curve	Radius 22m
Phase 9	Straight Line	Length 64m
Phase 10	90° curve	Length 17.4m

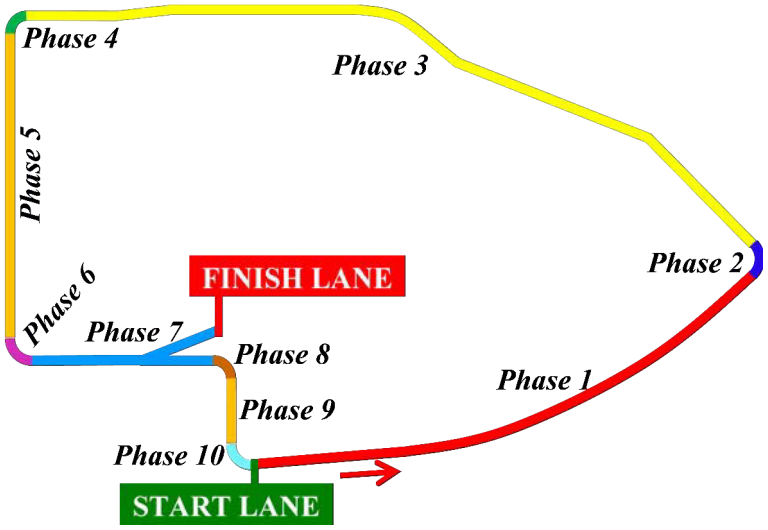


Figure 2.7: Consecutive phases in the Rotterdam's Ahoy circuit.

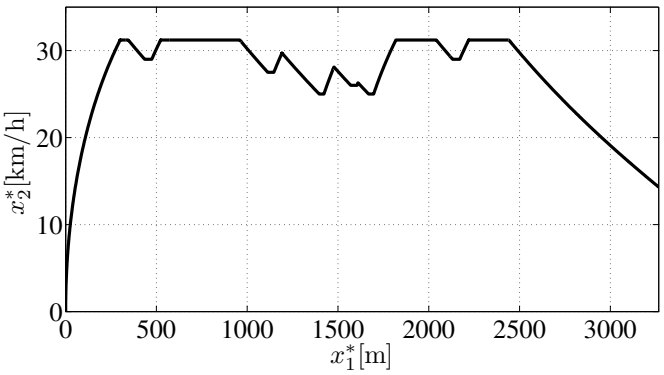
The optimal solution obtained with the GPOPS Toolbox$^{\text{TM}}$ to run two laps in the Rotterdam's Ahoy circuit, is depicted in Fig. 2.8. The total distance is 3.266km and the maximum allowed time is 468s, being $I_{batt_{max}} = 7$A. The total performance of the solution is 474.0826km/kWh, equivalent to run 4215km with one litre of fuel, according to (1.1).

2.4 Nonlinear discrete-time model

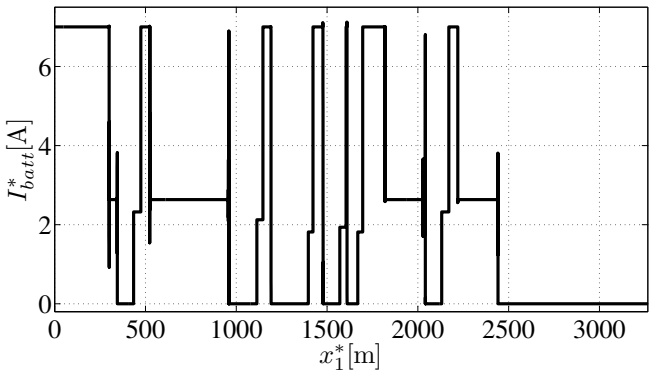
Having in mind the design of a control law which will be embedded on-board in a digital device, a discrete-time model must be obtained from the dynamics (2.6) and (2.7).

The Euler's Forward difference numerical approximation is well suited to perform the discretization of nonlinear dynamics [33]. In the Euler's Forward approximation the state vector $x(k+1)$ is written in an explicit way from the expression

$$\frac{dx}{dt}(k) = \frac{x(k+1) - x(k)}{T_s}, \tag{2.18}$$



(a)



(b)

Figure 2.8: Optimal driving strategy with performance of 474.0826km/kWh. (a) Optimal velocity profile. (b) Optimal battery current profile.

where T_s[s] is the sample time. It follows that

$$x(k+1) = T_s \frac{dx}{dt}(k) + x(k). \tag{2.19}$$

On the other hand, in the Tustin's method or in the Euler's backward discretization method, the expression $x(k+1)$ is derived in an implicit way

$$\text{Backward differentiation Euler's Method: } \frac{dx}{dt}(k) = \frac{x(k) - x(k-1)}{T_s}. \tag{2.20}$$

The implicit approaches are less amenable to find a discrete-time model of a

nonlinear dynamics, because they require the resolution of a nonlinear differential equation [33]. Hence, the Euler's Forward approximation will be used in the following to find a discrete-time representation of the nonlinear continuous-time vehicle dynamics.

From (2.6), (2.7) and (2.18), the following nonlinear discrete-time dynamic is obtained

$$x_1(k+1) = x_1(k) + T_s x_2(k), \tag{2.21}$$

and

$$x_2(k+1) = x_2(k) + T_s \frac{\eta k_t g_r}{m r_w} I_{batt}(k) - \frac{1}{2m} T_s \rho C_d A_f x_2(k)^2 - T_s g C_r \cos(\theta(k)) - T_s g \sin(\theta(k)). \tag{2.22}$$

When the discretization process is implemented, a zero holder is usually considered and introduces a delay of $T_s/2$ (see Fig. 2.9). This delay can influence the stability of the system if it is too large. In [33], it is demonstrated that the effect of the time delay over the system stability is tolerable if the time delay $T_s/2$ is smaller than one tenth of the rise time t_r (63% rise time rule for the velocity of operation) of the original continuous-time system, i.e,

$$\begin{aligned} \frac{T_s}{2} &\leq \frac{t_r}{10}, \\ T_s &\leq \frac{t_r}{5}. \end{aligned} \tag{2.23}$$

Therefore, for the system (2.21)-(2.22) the sampling time will be set as $T_s = 0.2$s.

2.5 Real-time tracking of the optimal driving strategy

In order to track the driving strategy, a control law must be designed. The control law must cope with the real-time constraints inherent to the digital devices wherein the control law will be embedded. The control law is appropriately

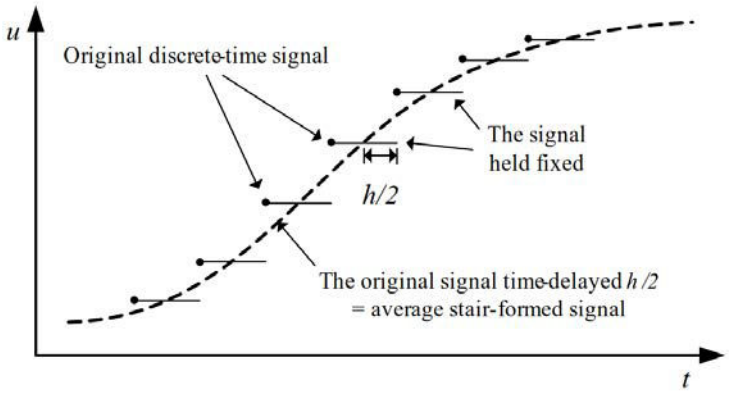


Figure 2.9: Time delay of the discretization process [33]. $h = T_s$ is the time step.

developed in Chapters 3-5. However, before proceeding further, the tracking problem must be properly introduced. For this purpose, the tracking error is first derived from the nonlinear dynamics of the vehicle. And then, two different representations of the tracking error are introduced as amenable models to achieve the design of the control law: Linearised model (Section 2.5.1) and LPV model (Section 2.5.2). The control law must steer to zero the tracking error in order to track the optimal reference, that is the driving strategy.

Consider the state $x^*(k) = [x_1^*(k),\, x_2^*(k)]^T \in \mathbb{R}^2$ and the input $I_{batt}^*(k)$, both of them obtained respectively from the sampling, with sampling period T_s, of the optimal trajectory $x^*(t) = [x_1^*(t),\, x_2^*(t)]^T \in \mathbb{R}^2$ and the sampling of the optimal input $I_{batt}^*(t)$. Recall that $(x_1^*(t), x_2^*(t), I_{batt}^*(t))$ are solution of (2.15) presented in Section 2.3. From the nonlinear discrete state $x(k) = [x_1(k),\, x_2(k)]^T \in \mathbb{R}^2$ of the vehicle, and the optimal one $x^*(k) = [x_1^*(k),\, x_2^*(k)]^T$, the following tracking error is introduced for every sampling time

$$\Delta x(k) = \begin{bmatrix} \Delta x_1(k) \\ \Delta x_2(k) \end{bmatrix} = \begin{bmatrix} x_1(k) - x_1^*(k) \\ x_2(k) - x_2^*(k) \end{bmatrix}. \tag{2.24}$$

Define $\Delta I_{batt}(k) = I_{batt}(k) - I_{batt}^*(k)$ as the difference between the actual input $I_{batt}(k)$ of the dynamics (2.22) and the sampled target control $I_{batt}^*(k)$ at time k.

Since the optimal references $x^*(k) = [x_1^*(k),\, x_2^*(k)]^T$ and $I_{batt}^*(k)$ follows the

dynamics (2.21) and (2.22), in virtue of the optimization problem (2.15), then the dynamics of the tracking error $\Delta x(k+1)$ reads

$$\Delta x(k+1) = \begin{bmatrix} \Delta x_1(k+1) \\ \Delta x_2(k+1) \end{bmatrix}, \tag{2.25}$$

where

$$\begin{aligned} \Delta x_1(k+1) &= x_1(k+1) - x_1^*(k+1), \\ &= x_1(k) + T_s x_2(k) - (x_1^*(k) + T_s x_2^*(k)), \\ &= \Delta x_1(k) + T_s \Delta x_2(k), \end{aligned} \tag{2.26}$$

and

$$\begin{aligned} &\Delta x_2(k+1) = x_2(k+1) - x_2^*(k+1), \\ &= x_2(k) + T_s \frac{\eta k_t g_r}{m r_w} I_{batt}(k) - \frac{1}{2m} T_s \rho C_d A_f x_2(k)^2 - T_s g C_r \cos(\theta(k)) - T_s g \sin(\theta(k)) \\ &- \left(x_2^*(k) + T_s \frac{\eta k_t g_r}{m r_w} I_{batt}^*(k) - \frac{1}{2m} T_s \rho C_d A_f x_2^*(k)^2 - T_s g C_r \cos(\theta(k)) - T_s g \sin(\theta(k)) \right), \\ &= (x_2(k) - x_2^*(k)) - \frac{1}{2m} T_s \rho C_d A_f \left(x_2(k)^2 - x_2^*(k)^2 \right) + T_s \frac{\eta k_t g_r}{m r_w} \left(I_{batt}(k) - I_{batt}^*(k) \right), \\ &= \Delta x_2(k) - \frac{1}{2m} T_s \rho C_d A_f \left(x_2(k) + x_2^*(k) \right) \Delta x_2(k) + T_s \frac{\eta k_t g_r}{m r_w} \Delta I_{batt}(k). \end{aligned} \tag{2.27}$$

Thus,

$$\Delta x(k+1) = \begin{bmatrix} \Delta x_1(k) + T_s \Delta x_2(k) \\ \Delta x_2(k) - \frac{1}{2m} T_s \rho C_d A_f \left(x_2(k) + x_2^*(k) \right) \Delta x_2(k) + T_s \frac{\eta k_t g_r}{m r_w} \Delta I_{batt}(k) \end{bmatrix}. \tag{2.28}$$

Notice that (2.28) is obtained since $\theta(k)$ is known for each time k and consequently is the same quantity involved in the optimal solution and in the actual dynamics. Conversely, $x_2(k)$ is not necessarily equal to $x_2^*(k)$ since $x_2(k)$ is the actual velocity and $x_2^*(k)$ is the target velocity at time k.

The control objective is to steer to zero the nonlinear tracking error $\Delta x_2(k+1)$. The quantity $\Delta x_1(k) = 0$ for all k, since $x_1(k)$ and $x_1^*(k)$ are the actual position

of the vehicle in the track. The dynamics involved in the design of the control is (2.28). In order to steer to zero (2.28), two approaches are proposed. The first one, is based on the linearisation of (2.28) around an operating point. In the second one, (2.28) is rewritten in the form of a Linear Parametric Varying (LPV) model to preserve the nonlinear features of (2.28).

2.5.1 Linearised model

The nonlinear dynamics (2.28) can be approximated by a Taylor series approximation around the steady-state operating point (x_{2_e}, I_{batt_e}) obtained from the solution of $x_2(k+1) = x_2(k)$ in (2.22), i.e.

$$I_{batt_e} = \frac{\rho C_d A_f r_w (x_{2_e})^2 + 2mgr_w C_r}{2\eta k_t g_r}, \tag{2.29}$$

where x_{2_e} is the average velocity of the optimal solution of (2.15).

Define the auxiliary function $f(x_2(k), x_2^*(k), I_{batt}(k), I_{batt}^*(k))$ in (2.27) as

$$\begin{aligned}\Delta x_2(k+1) &= f(x_2(k), x_2^*(k), I_{batt}, I_{batt}^*(k)) \\ &= (x_2(k) - x_2^*(k)) - \frac{1}{2m} T_s \rho C_d A_f \left(x_2(k)^2 - x_2^*(k)^2\right) + T_s \frac{\eta k_t g_r}{m r_w} \left(I_{batt}(k) - I_{batt}^*(k)\right).\end{aligned} \tag{2.30}$$

A linear approximation of (2.30), around the steady state (x_{2_e}, I_{batt_e}), can be found by developing the expansion of the Taylor series over $f(x_2(k), x_2^*(k), I_{batt}(k), I_{batt}^*(k))$ up to the order one, as follows

$$\begin{aligned}\Delta x_2(k+1) \approx & f(\cdot)\Big|_{\substack{x_2=x_{2_e} \\ x_2^*=x_{2_e} \\ I_{batt}=I_{batt_e} \\ I_{batt}^*=I_{batt_e}}} + \frac{\partial f}{\partial x_2}\Big|_{\substack{x_2=x_{2_e} \\ x_2^*=x_{2_e} \\ I_{batt}=I_{batt_e} \\ I_{batt}^*=I_{batt_e}}} (x_2 - x_{2_e}) + \frac{\partial f}{\partial x_2^*}\Big|_{\substack{x_2=x_{2_e} \\ x_2^*=x_{2_e} \\ I_{batt}=I_{batt_e} \\ I_{batt}^*=I_{batt_e}}} (x_2^* - x_{2_e}) \\ & + \frac{\partial f}{\partial I_{batt}}\Big|_{\substack{x_2=x_{2_e} \\ x_2^*=x_{2_e} \\ I_{batt}=I_{batt_e} \\ I_{batt}^*=I_{batt_e}}} (x_2 - x_{2_e}) + \frac{\partial f}{\partial I_{batt}^*}\Big|_{\substack{x_2=x_{2_e} \\ x_2^*=x_{2_e} \\ I_{batt}=I_{batt_e} \\ I_{batt}^*=I_{batt_e}}} (x_2^* - x_{2_e}).\end{aligned} \tag{2.31}$$

After proper mathematical developments, the expression (2.31) becomes

$$\Delta x_2(k+1) = \Delta x_2(k)\left(1 - \frac{T_s \rho C_d A_f}{m} x_{2_e}\right) + \frac{T_s \eta k_t g_r}{m r_w} \Delta I_{batt}(k). \tag{2.32}$$

Finally, the following linearised state-space representation is obtained for the tracking error (2.28)

$$\Delta x(k+1) = \mathbf{A}\Delta x(k) + \mathbf{B}\Delta I_{batt}(k), \tag{2.33}$$

where $\Delta x(k) = [x_1(k) - x_1^*(k),\ x_2(k) - x_2^*(k)]^T$ and $\Delta I_{batt}(k) = I_{batt}(k) - I_{batt}^*(k)$. The matrices $\mathbf{A}$ and $\mathbf{B}$ are given by

$$\mathbf{A} = \begin{bmatrix} 1 & T_s \\ 0 & 1 - \frac{T_s \rho C_d A_f}{m} x_{2_e} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{T_s \eta k_t g_r}{m r_w} \end{bmatrix}. \tag{2.34}$$

The full state is assumed to be accessible, which means that both the position and the velocity are measurable and the optimal reference of the position and the velocity are always available.

2.5.2 Linear Parametric Varying model

Alternatively to the linearisation around an operation point, the nonlinear tracking error (2.28) can also be expressed in the form of a Linear Parametric Varying (LPV) representation. Unlike the linearisation around an operation point, the LPV representation is a exact representation of the nonlinear model and thus preserves the nonlinear properties of the system. Additionally, it is motivated by the fact that it benefits from efficient tools, such as Linear Matrix Inequalities (LMIs), for the synthesis of the control law.

In general, a discrete-time LPV system is a system of the form

$$x(k+1) = \mathbf{A}(\lambda(k))x(k) + \mathbf{B}(\lambda(k))u(k). \tag{2.35}$$

$\lambda(k)$ is called the time-varying parameter. If $\lambda(k) \in \Phi$, for all k, with Φ a convex polytope, then (2.35) is a LPV polytopic system. In that case, the realization

matrices of the system (2.35) verify

$$[\mathbf{A}(\lambda(k))\ \mathbf{B}(\lambda(k))] \in \mathcal{C},\ \forall k, \tag{2.36}$$

being $\mathcal{C}$ the convex polytope

$$\mathcal{C} = Co\{[\mathbf{A}_1\ \mathbf{B}_1], \cdots, [\mathbf{A}_{p_\mathcal{C}}\ \mathbf{B}_{p_\mathcal{C}}]\}, \tag{2.37}$$

where Co denotes the convex hull, $[\mathbf{A}_j\ \mathbf{B}_j]$ the vertices of the convex hull for $1 \leq j \leq p_\mathcal{C}$ and $p_\mathcal{C}$ the number of vertices in $\mathcal{C}$. Any $[\mathbf{A}(\lambda(k))\ \mathbf{B}(\lambda(k))]$ within the convex polytope $\mathcal{C}$ is described by the following linear combination

$$[\mathbf{A}(\lambda(k))\ \mathbf{B}(\lambda(k))] = \sum_{j=1}^{p_\mathcal{C}} f_j(\lambda(k))[\mathbf{A}_j\ \mathbf{B}_j], \tag{2.38}$$

with

$$\sum_{j=1}^{p_\mathcal{C}} f_j(\lambda(k)) = 1,\ 0 \leq f_j(\lambda(k)) \leq 1. \tag{2.39}$$

The main requirements to find a LPV representation of a nonlinear system are (see [21])

- The parameter is a function of the states, i.e. $\lambda(k) = \lambda(x(k))$, and this function is known.
- The parameter is accessible on-line.

Following the prerequisites mentioned above, an equivalent LPV representation to the nonlinear dynamics (2.28) can be obtained. To that purpose, the variable $\lambda(k) = (x_2(k) + x_2^*(k))$ is introduced as the time-varying parameter accessible at each time k. Since x_2 and x_2^* are bounded, then the parameter $\lambda(k)$ is bounded as well, and $\lambda_{min} \leq \lambda(k) \leq \lambda_{max}$ for all k. Notice that the future behaviour of $\lambda(k)$ is not necessarily known but $\lambda(k)$ is always accessible.

The LPV representation of the nonlinear dynamics of the tracking error (2.28) is

$$\Delta x(k+1) = \mathbf{A}(\lambda(k))\Delta x(k) + \mathbf{B}\Delta I_{batt}(k), \tag{2.40}$$

with

$$\mathbf{A}(\lambda(k)) = \begin{bmatrix} 1 & T_s \\ 0 & 1 - \frac{1}{2m} T_s \rho C_d A_f \lambda(k) \end{bmatrix}, \text{ and } \mathbf{B} = \begin{bmatrix} 0 \\ T_s \frac{\eta k_t g_r}{m r_w} \end{bmatrix}. \tag{2.41}$$

From (2.38), the matrix $\mathbf{A}(\lambda(k))$ in (2.40) reads

$$\mathbf{A}(\lambda(k)) = f_1(\lambda(k))\mathbf{A}_1 + f_2(\lambda(k))\mathbf{A}_2, \tag{2.42}$$

where

$$\sum_{j=1}^{2} f_j(\lambda(k)) = 1, \ 0 \leq f_j(\lambda(k)) \leq 1. \tag{2.43}$$

The matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are characterized by the vertices $(\lambda_{min}, \lambda_{max})$ as follows

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} 1 & T_s \\ 0 & 1 - \frac{1}{2m} T_s \rho C_d A_f \lambda_{min} \end{bmatrix}, \\ \mathbf{A}_2 &= \begin{bmatrix} 1 & T_s \\ 0 & 1 - \frac{1}{2m} T_s \rho C_d A_f \lambda_{max} \end{bmatrix}. \end{aligned} \tag{2.44}$$

The functions f_1 and f_2 are characterized by the points indicated in Table 2.3.

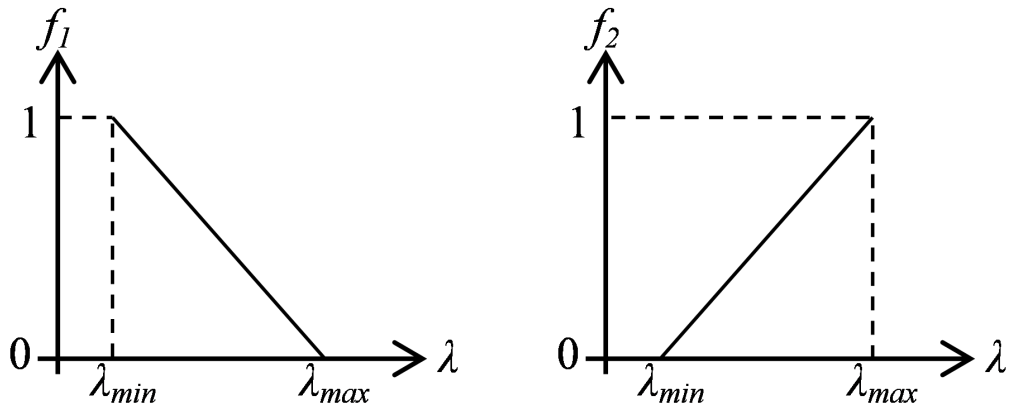
Table 2.3: Functions $f_1(\lambda(k))$ and $f_2(\lambda(k))$.

$\lambda(k)$	$f_1(\lambda(k))$	$f_2(\lambda(k))$
λ_{min}	1	0
λ_{max}	0	1

Therefore, f_1 and f_2 follows (see Fig. 2.10)

$$f_1(\lambda(k)) = \frac{\lambda_{max} - \lambda(k)}{\lambda_{max} - \lambda_{min}}, \text{ and } f_2(\lambda(k)) = \frac{\lambda(k) - \lambda_{min}}{\lambda_{max} - \lambda_{min}}. \tag{2.45}$$

Finally, any $[\mathbf{A}(\lambda(k)) \ \mathbf{B}]$ within the convex polytope $\mathcal{C}$ is described by the



(a) (b)

Figure 2.10: Functions of the vertices of the polytopic LPV representation. (a) $f_1(\lambda(k))$. (b) $f_2(\lambda(k))$.

following linear combination

$$[\mathbf{A}(\lambda(k))\ \mathbf{B}] = \sum_{j=1}^{2} f_j(\lambda(k))[\mathbf{A}_j\ \mathbf{B}]. \tag{2.46}$$

2.6 Benchmark

The efficiency and performances of the optimal driving strategy and the tracking strategies, are tested using a benchmark that has been built by the EMT. The benchmark emulates both the vehicle and the circuit profile. It is composed by four principal elements: the *Vir'volt* vehicle that has been equipped with several embedded electronics, an inertial cylinder that emulates the inertia of the vehicle and an electric motor that emulates frictions and disturbances, a microcontroller in which the tracking strategy has been embedded, and a graphical user interface used to visualize on-line the tracking performances. Those elements interact between them as is shown in the block diagram of Fig. 2.11. The main blocks are described in the following.

- The inertial cylinder and electric motor: The inertial cylinder, on which is placed the vehicle, allows to emulate the inertia of the vehicle. The inertial

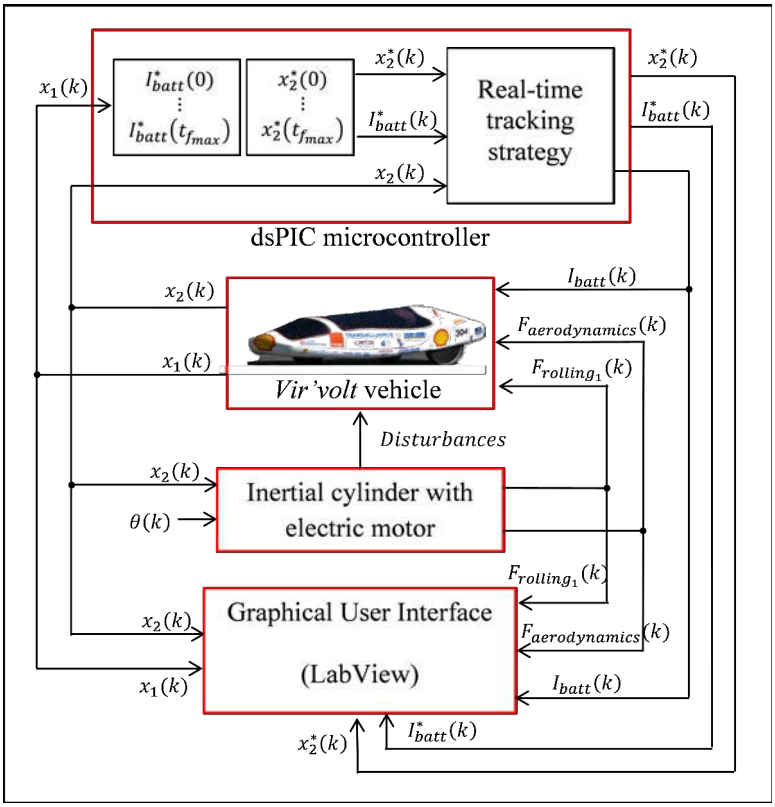


Figure 2.11: Benchmark ($F_{rolling_1} = F_{rolling} + F_{slope}$).

cylinder is in the same axis than an electric motor which emulates the friction forces such as the aerodynamic force $F_{aerodynamics}$, the rolling resistance $F_{rolling}$ and the slope resistance F_{slope} (see (2.1) in Section 2.2.1). The rolling resistance $F_{rolling}$ and the slope resistance F_{slope} depend of the slope θ of the road (see (2.3) and (2.4)). The slope is known all along the path. The aerodynamic resistance $F_{aerodynamics}$, depends on the square of the actual vehicle velocity $x_2(k)^2$ (see (2.2)). To properly emulate $F_{rolling}$, F_{slope} and $F_{aerodynamics}$ according to $\theta(k)$ and $x_2(k)^2$ at time k, the electric motor is regulated by a PID controller. The PID controller makes the electric motor to oppose to the movement of the vehicle in a quantity that corresponds to $F_{rolling}$, F_{slope} and $F_{aerodynamics}$, according to the actual measure of $\theta(k)$ and $x_2(k)^2$ (see Fig. 2.11). The electric motor is also equipped with an electric brake to emulate wind and others disturbances. In Fig. 2.12, the inertial

cylinder and the electric motor are presented.

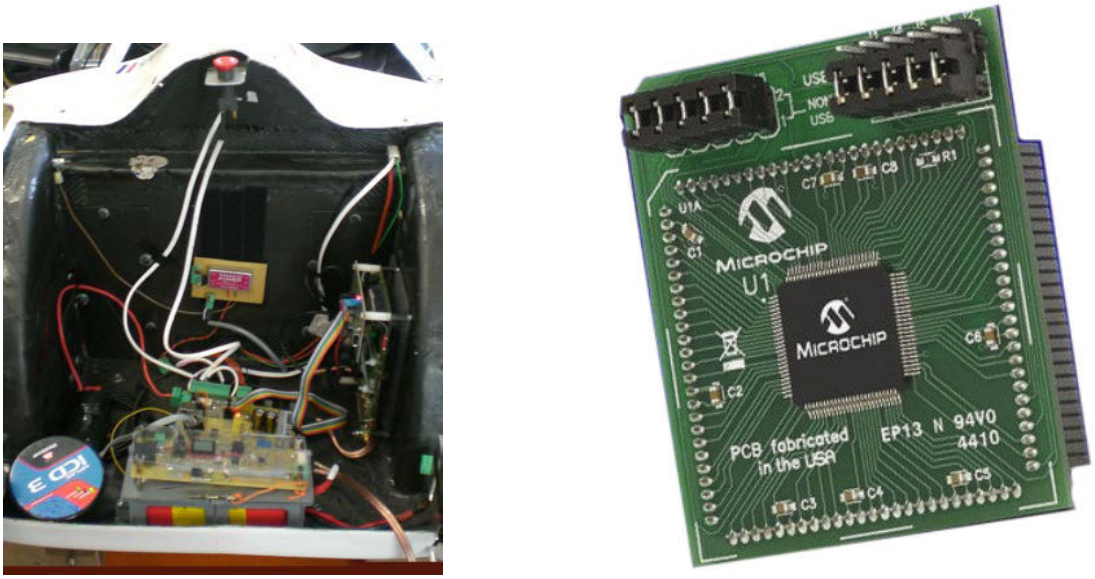
(a) (b) (c)

Figure 2.12: Benchmark in the year 2014. (a) *Vir'Volt* vehicle and benchmark. (b) Detail of the Benchmark. (c) Inertial cylinder and electric motor.

- The *Vir'volt* vehicle: the vehicle lies over the inertial cylinder and is submitted to disturbances, frictions, and other phenomena that may be present during the actual race. The vehicle is equipped with a microcontroller in which the control strategy has been embedded. The vehicle is also equipped with multiple sensors that measure the virtual travelled distance, the velocity of the vehicle and the current of the battery (see Fig. 2.13a). The travelled distance is considered as the actual position of the vehicle in the track. Those measures are used by the microcontroller to compute the control law (see Fig. 2.11).

- The microcontroller: A Microchip® dsPIC33EP512MU810 microcontroller (see Fig. 2.13b), is implemented on-board of the *Vir'volt* vechicle. This device has an internal clock of 7.37 MHz, a RAM with 53000 bytes capacity and a faster DMA RAM with 4096 bytes capacity. Its ADC samples at 500 kHz for 12 bits resolution and at 1 MHz for 10 bits resolution. The control strategy designed to track to the optimal driving strategy is embedded in the dsPIC. The optimal driving strategy $(x_1^*(k), x_2^*(k), I_{batt}^*(k))$ is saved in the memory of the dsPIC. On-line, the actual travelled distance $x_1(k)$ is used to find the corresponding reference velocity $x_2^*(k)$ according to the

position of the vehicle in the track. The reference velocity $x_2^*(k)$ is required by the embedded tracking strategy to compute the control law (see Fig. 2.11). In Fig. 2.13a are shown the embedded electronics implemented in the back of the vehicle.



(a) (b)

Figure 2.13: On-board implemented electronics. (a) Electronics implemented in the back of the vehicle. dsPic, power converter, etc. (b) Microchip® dsPIC33EP512MU810 microcontroller.

- The graphical user interface: a graphical user interface has been design in LabView™ to visualize the on-line performance of the tracking strategy. Data such as the references $x_2^*(k)$, $I_{batt}^*(k)$, and the control law $I_{batt}(k)$ are acquired using a RS-232 protocol from the dsPIC. The rest of the data such as $x_1(k)$, $x_2(k)$ and the frictional forces are collected using a data acquisition system (DAQ) that consists in a National Instruments analog DAQ device (USB-6289) (see Fig. 2.11).

The dynamics of the vehicle in the benchmark is a scaled version of the dynamics (2.6) and (2.7). The dynamics of the *Vir'volt* vehicle over the benchmark,

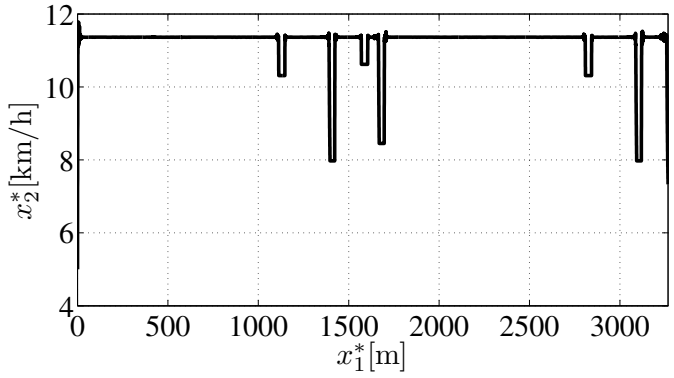
for flat path, is given by

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t), \\ \frac{dx_2(t)}{dt} &= 1.1228 I_{batt}(t) - 0.1125 x_2(t)^2 - 0.1893. \end{aligned} \tag{2.47}$$

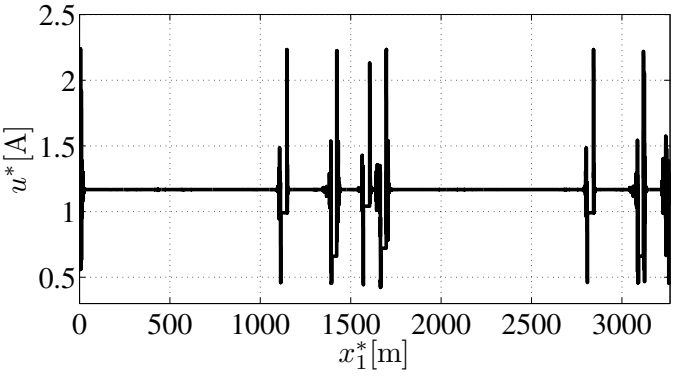
The discrete-time model (2.21) and (2.22) is given by

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k), \\ x_2(k+1) &= x_2(k) + 1.1228 T_s I_{batt}(k) - 0.1125 T_s x_2(k)^2 - 0.1893 T_s. \end{aligned} \tag{2.48}$$

In order to asses the performance of the tracking strategy in the benchmark, a optimal driving strategy is found for the dynamics (2.47), using the features of the Rotterdam's Ahoy track. The optimal driving strategy is computed for a run of two laps in the Ahoy track, i.e. $x_{1_{total}} = 3266$m, and a final maximal time $t_{f_{max}} = 1050$s. The solution of the optimal problem (2.15) for the dynamics (2.47) is depicted in Fig. 2.14.



(a)



(b)

Figure 2.14: Optimal driving strategy for the vehicle in the benchmark with performance of 408.4401km/kWh. (a) Optimal velocity profile. (b) Optimal battery current profile.

2.7 Conclusions

A model of the vehicle dynamics has been obtained and the unknown parameters have been estimated using experimental data. The identified dynamics was used to derive an optimal driving strategy that will be tracked on-line during the driving task. The tracking problem was formulated using a linearised model around an operating point and a Linear Parametric Representation (LPV).

In Chapters 3 and 4, proper control laws to perform the tracking of the optimal reference are developed using the linearised representation of the tracking error given by (2.33).

In Chapter 5, a proper control law to perform the tracking of the optimal reference is developed using the LPV representation of the tracking error given by (2.40).

Chapter 3

Tracking Model Predictive Control

3.1 Introduction

Model Predictive Control (MPC) has been well-admitted for the past 50 years as a suitable solution to deal with multi-variable constrained control problems and robustness in presence of model uncertainties and noise [53, 11]. Its origins go back to the years 1976 and 1978, in particular with the works of Richalet in *Model Predictive Heuristic Control* and *Model Algorithm Control* [69, 23]. The MPC is characterized by its ability to include soft and hard constraints, to deal with multivariable dynamics, and to perform optimization on-line [69, 23].

Over the last 20 years, feasibility of the optimization problem and stability of the control law have been major issues. Regarding the feasibility, the inherent trade-off between finite-horizon and constraints of MPC-based techniques raise challenging problems [11, 23]. In particular, when it comes to MPC-based tracking, such a trade-off may prevent the reachability of the reference [45]. As far as stability is concerned, several approaches have been proposed to provide some guarantees. The consideration of invariant sets as terminal constraint is one of the most popular. Indeed, the strategies based on invariant sets allow to guarantee convergence towards the origin by implicitly extending the prediction horizon to the infinity without any substantial increasing of the on-line computational

cost. The computation of the invariant sets, which may be usually polytopes or ellipsoids, is usually performed off-line [30, 40, 17], which allows a real-time implementation.

In this chapter, background on the issue of MPC-based tracking for LTI discrete-time systems under time-invariant polytopic constraints on the input and/or the state is recalled. The procedure to design an invariant set suitable to be used in the MPC strategy as a terminal set is described in Section 3.3. The invariant set guarantees asymptotic stability of the control law. In Section 3.4, a numerical example of tracking subject to polytopic constraints is presented. Finally, in Section 3.5, the MPC-based tracking under time-invariant constraints on the input and on the state is applied to the *Vir'volt* vehicle.

3.2 Preliminaries

Consider the linear system with state-space description

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k), \tag{3.1}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^p$ is the control input, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the dynamical matrix and $\mathbf{B} \in \mathbb{R}^{n\times p}$ is the input matrix. The pair $(\mathbf{A}, \mathbf{B})$ is assumed to be stabilizable, i.e. $\exists K$ s.t. $(\mathbf{A} + \mathbf{B}K)$ is Schur stable. The state is assumed to be accessible, i.e. $x(k)$ is fully known at each time k. The system (3.1) is assumed to be subject to constraints on the input and the state, that is

$$\begin{aligned} x(k) &\in X, \ \forall k \geq 0, \\ u(k) &\in U, \ \forall k \geq 0, \end{aligned} \tag{3.2}$$

with X a convex and closed subset of $\mathbb{R}^n$ and U a convex and compact subset of $\mathbb{R}^p$. X and U are assumed to be polytopic sets.

Define the tracking errors $\Delta x(k) = x(k) - \bar{x} \in \mathbb{R}^n$ and $\Delta u(k) = u(k) - \bar{u} \in \mathbb{R}^p$ around the steady-state targets $\bar{x} \in \mathbb{R}^n$ and $\bar{u} \in \mathbb{R}^p$ ($\bar{x}$ and $\bar{u}$ are solutions of (3.1), i.e. $\bar{x} = \mathbf{A}\bar{x} + \mathbf{B}\bar{u}$). The dynamics of the tracking error is given by

$$\Delta x(k+1) = \mathbf{A}\Delta x(k) + \mathbf{B}\Delta u(k), \tag{3.3}$$

with tracking constraints defined $\forall k \geq 0$ as

$$\begin{aligned} \Delta x(k) &\in X_\Delta, \\ \Delta u(k) &\in U_\Delta, \end{aligned} \tag{3.4}$$

where X_Δ is a convex and closed subset of $\mathbb{R}^n$ and U_Δ is a convex and compact subset of $\mathbb{R}^p$. Both subsets are assumed to be polytopic sets containing the origin in their interior, i.e. $0 \in int(X_\Delta)$ and $0 \in int(U_\Delta)$, hence $\bar{x} \in int(X)$ and $\bar{u} \in int(U)$ ($int(\cdot)$ denotes the interior of the set).

Remark 1 *The constraints X_Δ and U_Δ, are usually imposed by the application. If the constraints are too narrow, they may cause infeasibility of the problem if the performance requirements imposed to the control law are also too strict. Thus, a trade-off must be found between the constraints and the performance requirements. To that purpose, the constraints may be relaxed and/or the performances requirements of the control may be done less stringent. Hereafter, the constraints X_Δ and U_Δ are assumed to respect that trade-off.*

3.2.1 Polytopic constraints

The polytope X_Δ is described by

$$X_\Delta = \{\Delta x(k) \in \mathbb{R}^n : \mathbf{H}_{X_\Delta} \Delta x(k) \leq \bar{1}\}. \tag{3.5}$$

which is the hyperplane representation of the polytope X_Δ, with $\mathbf{H}_{X_\Delta} \in \mathbb{R}^{p_{X_\Delta} \times n}$ and $\bar{1} \in \mathbb{R}^{p_{X_\Delta}}$ is the ones vector $\bar{1} = [1, ..., 1]^T$. The scalar p_{X_Δ} is the facets number of the polytope X_Δ, and corresponds to the number of rows in $\mathbf{H}_{X_\Delta}$. The inequalities in the set (3.5) can be developed as

$$\begin{bmatrix} \mathbf{H}_{X_\Delta}(1,1) & \mathbf{H}_{X_\Delta}(1,2) & \cdots & \mathbf{H}_{X_\Delta}(1,n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{X_\Delta}(p_{X_\Delta},1) & \mathbf{H}_{X_\Delta}(p_{X_\Delta},2) & \cdots & \mathbf{H}_{X_\Delta}(p_{X_\Delta},n) \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{3.6}$$

Each *i-th* row in $\mathbf{H}_{X_\Delta}$ defines the equation

$$\begin{bmatrix} \mathbf{H}_{X_\Delta}(i,1) & \mathbf{H}_{X_\Delta}(i,2) & \cdots & \mathbf{H}_{X_\Delta}(i,n) \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} = 1, \ i = 1, ..., p_{X_\Delta}, \tag{3.7}$$

which is the equation of the hyperplane that describes the *i-th* facet in X_Δ.

The polytope U_Δ is also represented by a hyperplane representation as

$$U_\Delta = \{\Delta u(k) \in \mathbb{R}^p : \mathbf{H}_{U_\Delta} \Delta u(k) \leq \bar{1}\}, \tag{3.8}$$

with $\mathbf{H}_{U_\Delta} \in \mathbb{R}^{p_{U_\Delta} \times p}$ and $\bar{1} \in \mathbb{R}^{p_{U_\Delta}}$. The scalar p_{U_Δ} is the facets number of the polytope U_Δ. The scalars p_{X_Δ} and p_{U_Δ} are named the *complexity-index*, and are the number of facets, or equivalently the number of vertices, in the respective polytope X_Δ and U_Δ. Notice that polytopic representations are very handy to deal with linear constraints often encountered in practical applications [17].

3.2.2 Problem formulation

The aim of the MPC-based strategy is to enforce dynamics in (3.3) to reach the steady-state targets $\bar{x}$ and $\bar{u}$ while fulfilling the constraints (3.4). The control under concern here, consists in the on-line solution of the following infinite horizon open-loop optimization problem

$$\min_{\Delta u(k+i) \in \mathbb{R}^p, i=[0,\infty)} \left(J(k) = \sum_{i=0}^{\infty} \left(\Delta x^T(k+i) \mathbf{Q} \Delta x(k+i) + \Delta u^T(k+i) \mathbf{R} \Delta u(k+i) \right) \right)$$

$$\begin{aligned} s.t. \quad & \Delta x(k+i+1) = \mathbf{A}\Delta x(k+i) + \mathbf{B}\Delta u(k+i), && \forall i = 0, ..., \infty, \\ & \Delta x(k+i) \in X_\Delta, && \forall i = 0, ..., \infty, \\ & \Delta u(k+i) \in U_\Delta, && \forall i = 0, ..., \infty. \end{aligned} \tag{3.9}$$

The closed-loop control consists in applying at each execution time k the first element $\Delta u(k)$ of the optimal solution sequence $\Delta u(k), \Delta u(k+1), ..., \Delta u(\infty)$. The weighting matrices $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{p \times p}$ define the state and the input tracking costs respectively. Notice that the minimization in the problem (3.9) is

performed over a set of infinite dimension.

Notice that, in general, optimality does not imply stability [39]. However, as is pointed out in [53], asymptotic stability is assured for infinite horizon optimal control problems (under stabilizability and detectability). Thus, asymptotic stability is also assured for the infinite horizon optimization problem (3.9). Nevertheless, solving on-line infinite horizon problems is usually not practical (apart from standard H_2 and H_∞ control of linear systems) [53].

To tackle this problem, the performance criterion $J(k)$ in (3.9) is rewritten as

$$\begin{aligned}
J(k) &= \sum_{i=0}^{\infty} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i) \Big), \\
&= \sum_{i=0}^{N_p-1} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i) \Big) \\
&\qquad + \sum_{i=N_p}^{\infty} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i) \Big), \\
&= \sum_{i=0}^{N_p-1} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i) \Big) \\
&\qquad + \Delta x^T(k+N_p)\mathbf{P}\Delta x(k+N_p),
\end{aligned} \tag{3.10}$$

with $\mathbf{P} \in \mathbb{R}^{n\times n}$ the solution of the Riccati equation that solves the infinite-horizon LQR problem for the system (3.3), with weighting matrices $\mathbf{Q}$ and $\mathbf{R}$.

From (3.10), the problem (3.9) can been replaced by the following finite-

horizon open-loop problem

$$\min_{\Delta u(k+i)\in\mathbb{R}^p,\ i=[0,k+N_p-1]} \sum_{i=0}^{N_p-1} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i)\Big) + \Delta x^T(k+N_p)\mathbf{P}\Delta x(k+N_p),$$

$$\begin{aligned} s.t.\quad & \Delta x(k+i+1) = \mathbf{A}\Delta x(k+i) + \mathbf{B}\Delta u(k+i), && \forall i = 0,\dots,N_p-1,\\ & \Delta x(k+i) \in X_\Delta, && \forall i = 0,\dots,N_p-1,\\ & \Delta u(k+i) \in U_\Delta, && \forall i = 0,\dots,N_p-1,\\ & \Delta x(k+N_p) \in X_f, \end{aligned} \tag{3.11}$$

with N_p the prediction horizon and X_f the terminal (closed) set of feasible final states (see Fig. 3.1). The matrix $\mathbf{P}$ defines the terminal cost. The closed-loop control consists in applying at each execution time the first element $\Delta u(k)$ of the optimal solution sequence $\Delta u(k), \Delta u(k+1), ..., \Delta u(k+N_p-1)$. The solution of the problem (3.11) implies that after N_p steps, the state reaches the set X_f (see Fig. 3.1). In practice, only the first element $\Delta u(k)$ is applied and then (3.11) is solved again for every time k.

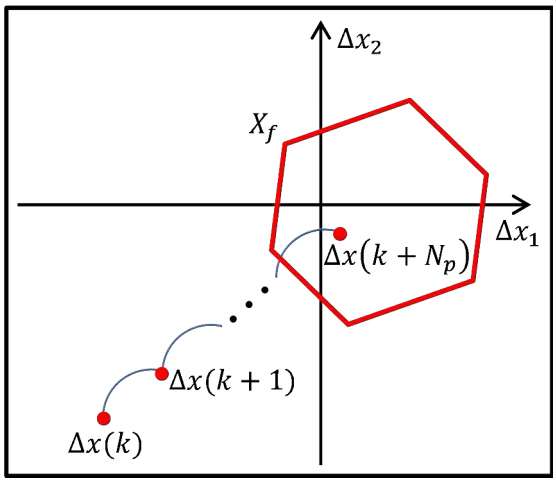


Figure 3.1: Set X_f of final states.

From [53, 11, 23], it has been well established that to enforce stability and convergence towards the origin, it is sufficient to meet two requirements:

- Make the terminal cost positive definite matrix $\mathbf{P}$ as the solution of the Riccati equation.
- Design the tracking terminal set X_f to be a controlled invariant set in the neighbourhood of the origin, under the LQR feedback gain denoted here with K.

Indeed, the constraint on the terminal set ensures that, after N_p steps, the predicted state reaches the terminal set. Since such a set is invariant, if the infinite horizon LQR control gain K is applied, the convergence towards the origin is guaranteed. Actually, the MPC consists in delivering, at each time k, the input $\Delta u(k)$, that is the first sample of the optimal input sequence resulting from the solution of (3.11). At time $k+1$, a new open-loop optimal control problem is solved. The following subsection recalls some background on the design of the terminal invariant set X_f.

3.3 Design of the invariant terminal set

The invariant sets can be ellipsoidal invariant sets or polytopic invariant sets. The polytopic invariant sets are less conservative than the ellipsoidal invariant set, but are harder to compute. Since the computation of the invariant set will be performed off-line, a less conservative polytopic invariant will be chosen.

The following procedure corresponds to the standard procedure (see [17] for example) to design a maximal polytopic invariant set, but is particularized for the tracking problem (3.11).

Consider the closed-loop system

$$\Delta x(k+1) = (\mathbf{A} + \mathbf{B}K)\Delta x(k), \tag{3.12}$$

obtained from (3.3) and from the stabilizing feedback control law $\Delta u(k) = K\Delta x(k)$. The gain $K \in \mathbb{R}^{p\times n}$ is derived from $\mathbf{P}$, that is the solution of the Riccati equation related to the infinite-horizon stabilization problem for system (3.3) with the quadratic cost weighting matrices $\mathbf{R}$ and $\mathbf{Q}$ involved in (3.11).

Define the closed-loop constraints for (3.12) as

$$\mathcal{X}_\Delta = \{\Delta x(k) \in \mathbb{R}^n : \Delta x(k) \in X_\Delta,\ K\Delta x(k) \in U_\Delta\}, \tag{3.13}$$

where X_Δ and U_Δ fulfil (3.4). The constraints (3.13) can be equivalently rewritten as

$$\mathcal{X}_\Delta = X_\Delta \cap X_{\Delta u}(U_\Delta), \tag{3.14}$$

where

$$X_{\Delta u}(U_\Delta) = \{\Delta x(k) \in \mathbb{R}^n : K\Delta x(k) \in U_\Delta\}. \tag{3.15}$$

Since the tracking constraints X_Δ and U_Δ in (3.4) are assumed to be polytopic, then the closed-loop constraint (3.13) can be expressed as a convex polytopic set containing the origin in its interior. As a result, $\mathcal{X}_\Delta$ can be described by a hyperplane representation with complexity-index $p_\mathcal{X} \in \mathbb{N}$ (i.e. $p_\mathcal{X}$ is the number facets of the polytope in the hyperplane representation):

$$\mathcal{X}_\Delta = \{\Delta x(k) \in \mathbb{R}^n : \mathbf{H}_{\mathcal{X}_\Delta}\Delta x(k) \leq \bar{1}\}, \tag{3.16}$$

where

$$\mathbf{H}_{\mathcal{X}_\Delta} = \begin{bmatrix} \mathbf{H}_{X_\Delta} \\ \mathbf{H}_{U_\Delta}K \end{bmatrix}, \tag{3.17}$$

with $\mathbf{H}_{\mathcal{X}_\Delta} \in \mathbb{R}^{p_\mathcal{X} \times n}$, $\bar{1} \in \mathbb{R}^{p_\mathcal{X}}$ being the ones vector $\bar{1} = [1, ..., 1]^T$ and $p_\mathcal{X} \leq p_{X_\Delta} + p_{U_\Delta}$. The matrices $\mathbf{H}_{X_\Delta} \in \mathbb{R}^{p_{X_\Delta} \times n}$ and $\mathbf{H}_{U_\Delta} \in \mathbb{R}^{p_{U_\Delta} \times p}$ are the matrices of the hyperplane representation of the tracking constraints X_Δ and U_Δ, respectively, with p_{X_Δ} and p_{U_Δ} the number of facets of each polytope.

Now, the definition of an invariant set for the LTI discrete-time system (3.12) is introduced. The definition is borrowed from the general definition given in [16] and [17].

Definition 1 *A closed and convex set $\Omega \subseteq \mathbb{R}^n$ with $0 \in int(\Omega)$ (that is Ω is a C-set), is said to be a positively invariant set for the system (3.12) under (tracking) constraints $\mathcal{X}_\Delta$ given in (3.13), if for all $\Delta x(k) \in \Omega$ then $\Delta x(k+1) = (\boldsymbol{A} + \boldsymbol{B}K)\Delta x(k) \in \Omega$, being $\Omega \subseteq \mathcal{X}_\Delta$.*

Otherwise stated, for all $\Delta x(k) \in \Omega$, that is for any state $\Delta x(k)$ which has

reached Ω, it holds that $\Delta x(k+i) \in \Omega$ for all $i \geq 0$, or equivalently that the state can no longer escape from Ω. The notion of *positively* refers to the fact that only the future states satisfy $\Delta x(k+i) \in \Omega$, that is to $i \geq 0$. Hereafter, the positively invariant sets will be merely called *invariant sets* for brevity since only the positive invariance will be considered.

Since $\mathcal{X}_\Delta$ is a bounded polytope with the origin in its interior and $\mathbf{A}+\mathbf{B}K$ is Schur stable, then the maximal invariant set Ω in $\mathcal{X}_\Delta$ is also a convex polytope [30]. For convenience, the set Ω is considered to be described by its vertex representation

$$\Omega = \left\{ \Delta x(k) \in \mathbb{R}^n : \Delta x(k) = \mathbf{V}_\Omega \mathbf{w},\ \sum_{i=1}^{p_\Omega} \mathbf{w}(i) = 1,\ \mathbf{w}(i) \geq 0,\ \forall i \right\}, \tag{3.18}$$

where $\mathbf{V}_\Omega \in \mathbb{R}^{n \times p_\Omega}$ is the matrix whose columns are the vertices of Ω, $\mathbf{w} \in \mathbb{R}^{p_\Omega}$, and p_Ω is the complexity-index (number of vertices in the vertex representation) of the polytopic invariant set Ω. The equality in (3.18) can be rewritten as

$$\begin{bmatrix} \mathbf{V}_\Omega(1,1) & \mathbf{V}_\Omega(1,2) & \cdots & \mathbf{V}_\Omega(1,p_\Omega) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{V}_\Omega(n,1) & \mathbf{V}_\Omega(n,2) & \cdots & \mathbf{V}_\Omega(n,p_\Omega) \end{bmatrix} \begin{bmatrix} \mathbf{w}(1) \\ \mathbf{w}(2) \\ \vdots \\ \mathbf{w}(p_\Omega) \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix}. \tag{3.19}$$

Each *i-th* column in $\mathbf{V}_\Omega$ is the *i-th* vertex of the polytope Ω. Notice that if $\mathbf{w}(i) = 1,\ \mathbf{w}(j) = 0,\ j \neq i$, then

$$\begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} \mathbf{V}_\Omega(1,i) \\ \vdots \\ \mathbf{V}_\Omega(n,i) \end{bmatrix}, \tag{3.20}$$

where $[\mathbf{V}_\Omega(1,i),\ \cdots,\ \mathbf{V}_\Omega(n,i)]^T$ is the *i-th* vertex of Ω.

To determine Ω, one can resort to the *backward iterative algorithm* (see [17, 16] for further details). The *backward iterative algorithm* can be summed up as follows.

Algorithm 1

1. Set $k = 0$ and $\mathcal{X}_{\Delta_0} = \mathcal{X}_\Delta$ defined by (3.16).
2. Calculate $\mathcal{X}_{\Delta_{-k-1}}$ from $\mathcal{X}_{\Delta_{-k}}$ as given by

$$\mathcal{X}_{\Delta_{-k-1}} = \{\Delta x(x) \in \mathcal{X}_\Delta : (\mathbf{A} + \mathbf{B}K)\Delta x(x) \in \mathcal{X}_{\Delta_{-k}}\}. \tag{3.21}$$

3. Check whether $\mathcal{X}_{\Delta_{-k}} \subseteq (1+\epsilon)\mathcal{X}_{\Delta_{-k-1}}$, with ϵ a fixed numerical tolerance introduced for practical purposes.
4. If $\mathcal{X}_{\Delta_{-k}} \subseteq (1+\epsilon)\mathcal{X}_{\Delta_{-k-1}}$ holds, then $\mathcal{X}_{\Delta_{-k-1}}$ is an admissible invariant set for $\mathcal{X}$. If it does not hold, then move on time $k+1$ and repeat the process from Step 2 until $\mathcal{X}_{\Delta_{-k}} \subseteq (1+\epsilon)\mathcal{X}_{\Delta_{-k-1}}$ holds, or until a maximum allowed number of steps is reached (for the sake of finite accuracy implementation).

In other words, the pre-images of $\mathcal{X}_\Delta$ obtained from the inverse dynamics of (3.12) are successively computed back in time and trimmed to get the largest polytopic invariant set included in $\mathcal{X}_\Delta$. To clarify the MPC strategy and the *backward iterative algorithm*, an academic example is detailed in next section.

3.4 MPC-based tracking with time-invariant constraints: an academic example

Consider the system $x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k)$ with

$$\mathbf{A} = \begin{bmatrix} 0.9 & 0.25 \\ -0.25 & 0.9 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0.5 \\ 2 \end{bmatrix}, \tag{3.22}$$

and $x(k) = [x_1(k), x_2(k)]^T$. The aim is to derive a control law so that the state vector $x(k)$ tracks a given trajectory with tracking constraints X_Δ and U_Δ. The

reference is (see Fig. 3.2 and 3.3)

$$\bar{x}(k) = \begin{cases} [0.15; -0.08]^T & \text{for } k \leq 50, \\ [0; -0.03]^T & \text{for } 50 < k \leq 100, \\ [-0.1; 0.02]^T & \text{for } 100 < k \leq 150, \\ [0; -0.06]^T & \text{for } k > 150. \end{cases} \tag{3.23}$$

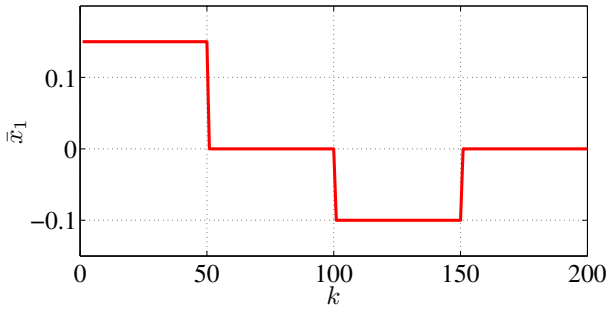


Figure 3.2: Reference $\bar{x}_1(k)$.

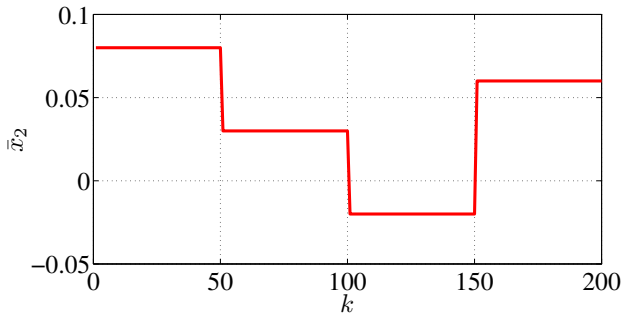


Figure 3.3: Reference $\bar{x}_2(k)$.

The tracking constraints are described by

$$X_\Delta = \{\Delta x(k) \in \mathbb{R}^n : \mathbf{H}_{X_\Delta} \Delta x(k) \leq \bar{1}\} \text{ and } U_\Delta = \{\Delta u(k) \in \mathbb{R}^p : \mathbf{H}_{U_\Delta} \Delta u(k) \leq \bar{1}\}, \tag{3.24}$$

where

$$\mathbf{H}_{X_\Delta} = \begin{bmatrix} \frac{1}{0.15} & 0 \\ 0 & \frac{1}{0.05} \\ \frac{-1}{0.2} & 0 \\ 0 & \frac{-1}{0.08} \end{bmatrix} = \begin{bmatrix} 6.6667 & 0 \\ 0 & 20 \\ -5 & 0 \\ 0 & -12.5 \end{bmatrix}, \; \mathbf{H}_{U_\Delta} = \begin{bmatrix} \frac{1}{0.01} \\ \frac{-1}{0.01} \end{bmatrix} = \begin{bmatrix} 100 \\ -100 \end{bmatrix}. \tag{3.25}$$

The constraints X_Δ corresponds to $-0.2 \leq \Delta x_1(k) \leq 0.15$ and $-0.08 \leq \Delta x_2(k) \leq 0.05$, for all k, and the constraints U_Δ to $-0.01 \leq \Delta u(k) \leq 0.01$, for all k. The prediction horizon is $N_p = 10$. The weighting matrices of the MPC problem (3.11) are $\mathbf{Q} = [1\ 0; 0\ 1]$ and $\mathbf{R} = 30$. The solution of the infinite-horizon problem is $\mathbf{P} = [4.6534\ 0.5613; 0.5613\ 3.0237]$ and the corresponding gain is $K = [-0.0343\ -0.1478]$.

3.4.1 Computation of the terminal invariant set

The matrix $\mathbf{H}_{\mathcal{X}_\Delta}$ defined as in (3.17), which characterizes the closed-loop system constraints $\mathcal{X}_\Delta$ given by (3.16), reads

$$\mathbf{H}_{\mathcal{X}_\Delta} = \begin{bmatrix} 6.6667 & 0 \\ 0 & 20 \\ -5 & 0 \\ 0 & -12.50 \\ -3.4304 & -14.7758 \\ 3.4304 & 14.7758 \end{bmatrix}, \tag{3.26}$$

with complexity-index (number of facets in the polytope) $p_\mathcal{X} = 6$. The polytopic constraints X_Δ and $\mathcal{X}_\Delta$ are depicted in Fig. 3.4. The figure clearly shows that $\mathcal{X}_\Delta$ is a subset of X_Δ, and it is in accordance with (3.14). The successive polytopes $\mathcal{X}_{\Delta_{-k-1}}$, involved in the *backward iterative algorithm* (see Algorithm 1), are depicted in red dotted lines in Fig. 3.5. The resulting invariant set Ω, that is the largest invariant set contained in $\mathcal{X}_\Delta$, is found after 5 steps and is portrayed in solid line in Fig. 3.5.

The resulting invariant set Ω is given by the following vertex representation,

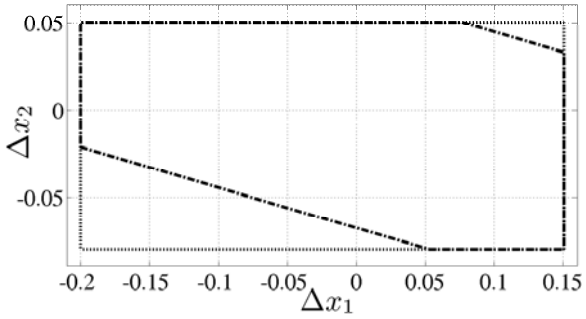


Figure 3.4: Polytopic constraints. Dashed line: constraints X_Δ. Dash-dotted line: constraints $\mathcal{X}_\Delta$.

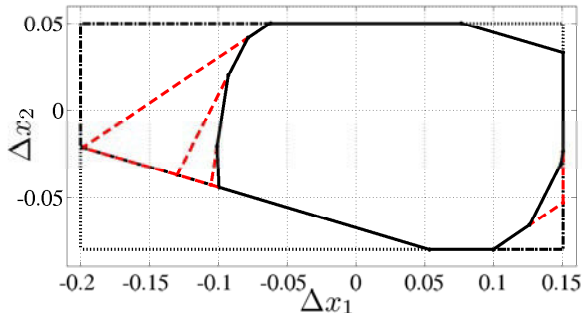


Figure 3.5: Backward iterative algorithm. Red dashed lines: pre-images of $\mathcal{X}_\Delta$. Solid line: final invariant set Ω.

the complexity-index (number of vertices in the polytope) being $p_\Omega = 12$:

$$\mathbf{V}_\Omega = \begin{bmatrix} 0.1500 & 0.0329 \\ 0.0993 & -0.0800 \\ -0.0928 & 0.0194 \\ 0.0531 & -0.0800 \\ 0.0761 & 0.0500 \\ -0.0621 & 0.0500 \\ -0.0785 & 0.0413 \\ -0.0996 & -0.0446 \\ -0.1010 & -0.0211 \\ 0.1257 & -0.0661 \\ 0.1485 & -0.0311 \\ 0.1500 & -0.0238 \end{bmatrix}^T . \tag{3.27}$$

3.4.2 Closed loop response

The problem (3.11), with polytopic invariant set given by (3.27), is solved for the dynamics (3.22) with initial conditions $x(0) = [0; 0]^T$. The results of the tracking are depicted in Fig. 3.6-3.10. The results show that the tracking is achieved and the constraints are fulfilled, as expected.

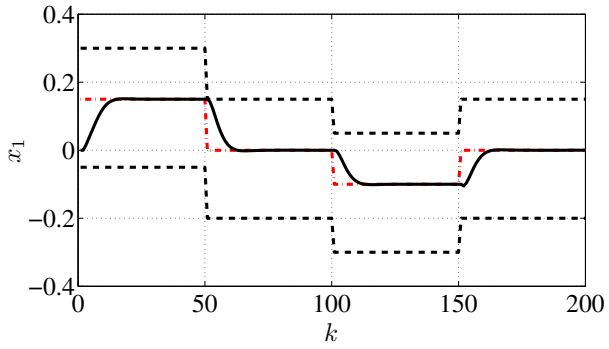


Figure 3.6: Tracking of $\bar{x}_1(k)$. Solid line: tracking response of $x_1(k)$. Red dash-dotted line: $\bar{x}_1(k)$. Dashed line: time-invariant constraints X_Δ for $\Delta x_1(k)$.

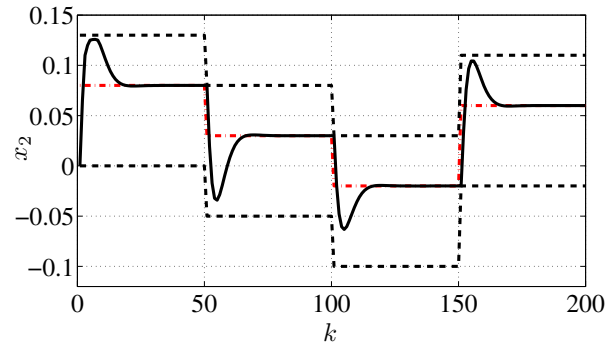


Figure 3.7: Tracking of $\bar{x}_2(k)$. Solid line: tracking response of $x_2(k)$. Red dash-dotted line: $\bar{x}_2(k)$. Dashed line: time-invariant constraints X_Δ for $\Delta x_2(k)$.

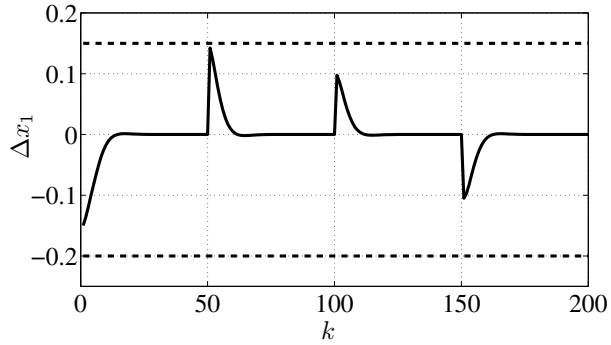


Figure 3.8: Tracking error $\Delta x_1(k)$. Solid line: tracking error $\Delta x_1(k)$. Dashed line: time-invariant constraints X_Δ for $\Delta x_1(k)$.

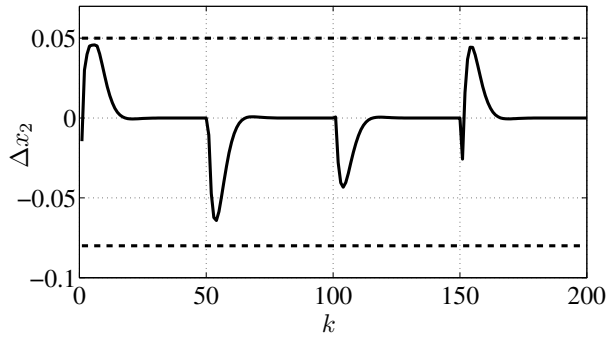


Figure 3.9: Tracking error $\Delta x_2(k)$. Solid line: tracking error $\Delta x_2(k)$. Dashed line: time-invariant constraints X_Δ for $\Delta x_2(k)$.

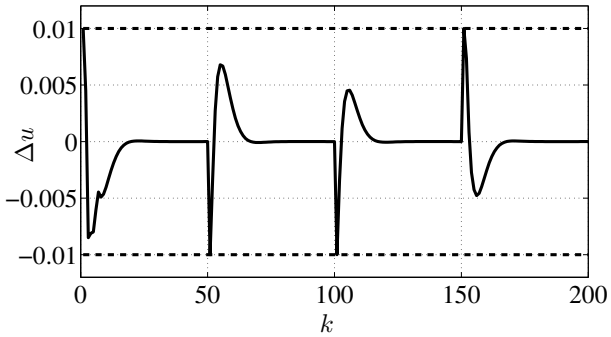


Figure 3.10: Tracking error $\Delta u(k)$. Solid line: tracking error $\Delta u(k)$. Dashed line: time-invariant constraints U_Δ.

3.5 MPC-based tracking with time-invariant constraints: application to the Vir'volt vehicle

Consider the linearised model (2.33) of the *Vir'volt* electric vehicle around the operating point (x_{2_e}, I_{batt_e}). In the following, the problem (3.11) is considered for the dynamics (2.33) with tracking reference given by the sampled solution of the optimization problem (2.15) that is depicted in Fig. 2.8. Hereafter, the battery current I_{batt} will be simply denoted with u, therefore $I^*_{batt} = u^*$ and $I_{batt_e} = u_e$.

The weighting matrices of the MPC problem (3.11) are $\mathbf{Q} = [0\ 0; 0\ 1]$ and $\mathbf{R} = 1$. The solution of the infinite-horizon problem is $\mathbf{P} = [0\ 0; 0\ 139.75]$ and the corresponding gain is $K = [0\ -0.6411]$. The constraints for $\Delta x_2(k)$ are imposed as $-2\text{km/h} \leq \Delta x_2(k) \leq 1\text{km/h}$ ($-0.56\text{m/s} \leq \Delta x_2(k) \leq 0.28\text{m/s}$). The tracking only concerns the velocity, thus the constraints for $\Delta x_1(k)$ are relaxed and set as $-10\text{m} \leq \Delta x_1(k) \leq 10\text{m}$. The constraints X_Δ, as in (3.5), are described by

$$\mathbf{H}_{X_\Delta} = \begin{bmatrix} \frac{1}{10} & 0 \\ 0 & \frac{1}{0.28} \\ \frac{-1}{10} & 0 \\ 0 & \frac{-1}{0.56} \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ 0 & 3.6 \\ -0.1 & 0 \\ 0 & -1.8 \end{bmatrix}. \tag{3.28}$$

The constraints U_Δ are set as $-0.5\text{A} \leq \Delta u(k) \leq 0.5\text{A}$. The polytope U_Δ is described by

$$\mathbf{H}_{U_\Delta} = \begin{bmatrix} \frac{1}{0.5} \\ \frac{-1}{0.5} \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}. \tag{3.29}$$

The constraints U_Δ are shown in Fig. 3.11.

The closed-loop constraints $\mathcal{X}_\Delta$ defined in (3.16) are depicted in Fig. 3.12. The *backward iterative algorithm* (see Algorithm 1) is applied to $\mathcal{X}_\Delta$ and the resulting terminal invariant set Ω is found after 11 steps. It is portrayed in Fig. 3.12. The complexity-index (number of vertices) of the polytope Ω is $p_\Omega = 6$. The polytope is described by

$$\mathbf{V}_\Omega = \begin{bmatrix} 10 & 10 & 9.4 & -8.8 & -10 & -10 \\ -0.55 & 0 & 0.27 & -0.55 & 0.27 & 0 \end{bmatrix}. \tag{3.30}$$

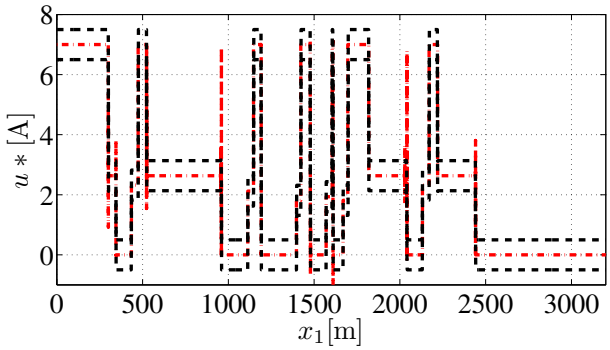


Figure 3.11: Optimal control $u^*(k)$ and tracking constraints. Red dash-dotted line: $u^*(k)$. Dashed line: constraints U_Δ.

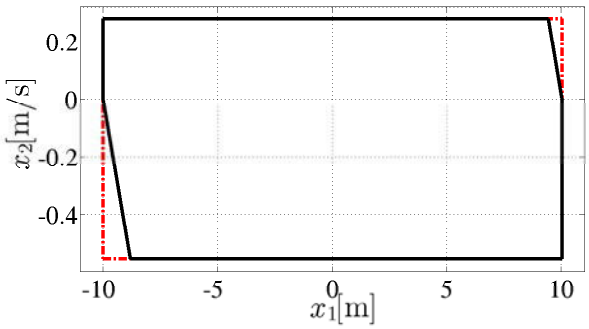


Figure 3.12: Maximal invariant set. Reed dash-dotted line: closed-loop constraints $\mathcal{X}_\Delta$. Solid line: final invariant set Ω.

The closed-loop control law (3.11) is applied to the nonlinear discrete-time dynamics of the vehicle, given by (2.21) and (2.22), according to the block diagram of Fig. 3.13. The operating point is fixed as the average velocity of the optimal solution depicted in Fig. 2.8, i.e. $x_{2_e} = 27$km/h. The vehicle is subject to a mass variation of 20% from its original value according to Table 2.1. The tracking of the optimal velocity is depicted in Fig. 3.14. In Fig. 3.15-3.16 are depicted $\Delta x_2(k)$ and $\Delta u(k)$, respectively. It can be observed that the tracking task is well achieved while the constraints are satisfied.

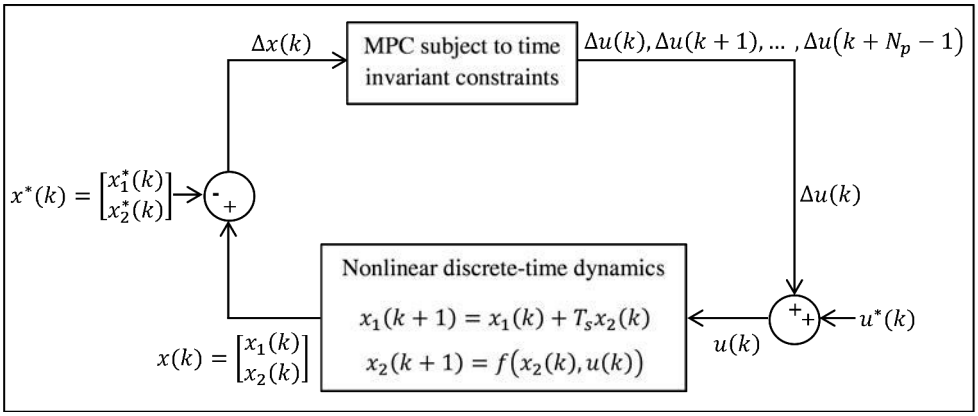


Figure 3.13: Closed-loop implementation of the MPC. The function that describes the velocity $x_2(k+1) = f(x_2(k), u(k))$ is given by (2.22).

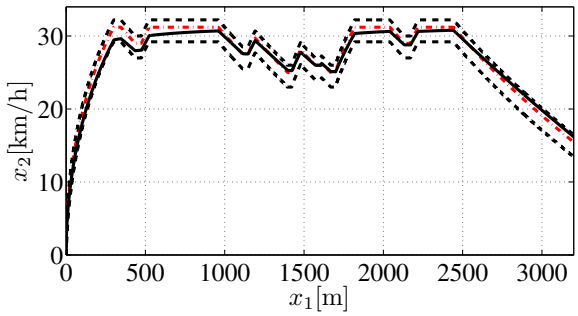


Figure 3.14: Tracking of the reference $x_2^*(k)$. Solid line: tracking response of $x_2(k)$ with a mass variation of 20%. Red dash-dotted line: reference. Dashed line: constraints X_Δ for $x_2(k)$.

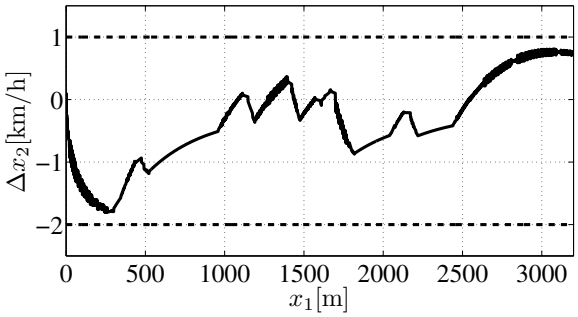


Figure 3.15: Tracking error $\Delta x_2(k)$. Solid line: tracking error $\Delta x_2(k)$. Dashed line: constraints X_Δ for $x_2(k)$.

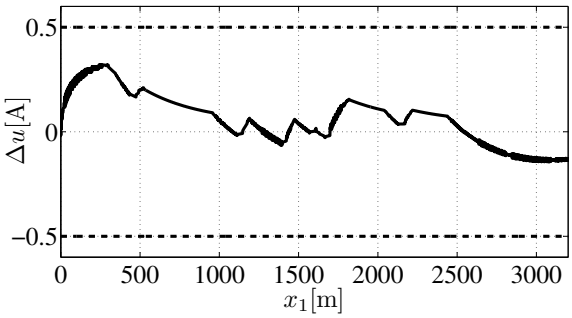


Figure 3.16: Tracking error $\Delta u(k)$. Solid line: tracking error $\Delta u(k)$. Dashed line: constraints U_Δ.

3.6 Conclusions

The issue of MPC-based tracking for LTI discrete-time systems under time-invariant constraints on the input and/or the state has been recalled. The use of a suitable invariant set as terminal set in the MPC problem has allowed to guarantee asymptotic stability to the reference. It has been noticed that the features of the systems, such as the dynamics and the constraints, are intimately involved in the design of the invariant set. The MPC strategy has been particularized to a tracking task. It has been illustrated on both an academic example and the linearised model of the *Vir'volt* vehicle.

The constant constraints (symmetric or asymmetric) may not be necessarily well suitable for all the stages of the driving strategy. In particular, for the *Vir'volt* vehicle, time-varying constraints may be more appropriated so that they are adapted to different stages of the tracking task.

Indeed, regarding the application of the control law to the low consumption *Vir'volt* vehicle, it can be observed that the discrepancies regarding the mass of the vehicle or the frictions are more critical during the initial acceleration and the final deceleration. The initial acceleration is intended to reach as soon as possible the reference to accomplish the constraint of the final time, and the final deceleration corresponds to turn off the motor to save energy. Thus, the initial acceleration and the final deceleration are expected to be always present (under feasibility conditions) in the solution of the problem (2.15). Therefore, particular constraints may be imposed only at the beginning and at the end of the tracking

of the velocity $x_2^*(k)$. This particular constraints are not necessarily the same that those required in the rest of the tracking task.

As a solution to deal with the problem stated above, it would be interesting to derive a MPC strategy that allows to incorporate time-varying constraints adapted to the features of the tracking task. The problem lies in that the proposed solution must be suited for low computational resources and real-time constraints. A solution to this problem is proposed in Chapter 4.

Chapter 4

Tracking under time-varying polytopic constraints

4.1 Introduction

The consideration of control problems with signals subject to time-varying constraints is often faced in real world applications. Despite the effervescence of works dealing with MPC, only few ones address the problem under time-varying constraints. As some exceptions, the following contributions can be quoted.

- In Bemporad et al. [12, 10], time-varying constraints that apply to the input and the output are incorporated in the *Explicit model predictive control*. In this approach, the time-varying constraints are required to be fully known a priori and the controlling sequences must be computed off-line for all the possible variations of the constraints. Hence, the implementation of the MPC requires a massive storage capacity.

- In the work of Mayne et al. [51], time-varying constraints have also been considered as an extension of the *output feedback model predictive control* in [52]. The approach in [51] is based on a stable state estimator and a robust tube-based MPC which is solved on-line. However, the existence and the knowledge of a common invariant set satisfying the states and inputs constraints for all the time instants is required.

- Wada et al. [67] address the feasibility and stability issues related to a MPC-based tracking problem for LTI systems with time-varying constraints, more specifically time-varying input saturation levels. In [67], on-line LMIs optimization problems are involved in the design of a *gain-scheduled feedback control law*, that is a feedback gain that depends on the time-varying saturation level of the input. However, solving on-line the LMIs may be unaffordable for real-time applications.

- Finally, it must be stressed that some well-admitted MPC-based tracking formulations can be very effective for time-invariant constraints but cannot be suited to cope with time-varying ones. It is precisely the case for the MPC-based tracking strategy presented in [45] which benefits from larger domain of attraction than the standard MPC formulation, but time-varying constraints are not allowed.

All in all, the consideration of time-varying constraints is still challenging when efficient MPC approaches compatible with real-time applications are required. In the following, a solution to the tracking problem for LTI systems subject to time-varying constraints is presented. The proposed solution is characterized by a control algorithm with low computational cost suitable to real-time applications. To enforce stability, the standard MPC-based tracking strategy with the terminal invariant set [53, 23, 62] is used. The problem of the standard approach lies in that the computation of the invariant set becomes intricate when considering time-varying constraints. Indeed, if the constraints change in time, the invariant set is also time-varying and so, must be recomputed on-line. And yet, when real-time applications are sought, it is far from being computationally affordable.

As a clue to tackle this problem, this chapter introduces an approach based on an homothetic transformation which consists in a contraction/dilatation of a predefined invariant set previously computed off-line. Such a transformation allows to fit on-line the time-varying constraints. The invariant set and the time-varying constraints both admit polytopic representations. The resulting parameter-dependent invariant set is an admissible terminal constraint for the MPC problem and guarantees asymptotic stability. Since the homothetic transformation is merely characterized by the dilating/contracting factor required to

fit the constraints, the computational cost for the on-line procedure boils down to the calculation of such a factor, and thus the MPC becomes suitable for real-time applications despite the time-varying constraints.

This chapter is organized as follows. In Section 4.2, the issue of MPC tracking problem subject to time-varying constraint is presented. The time-varying constraints are described using time-varying polytopic representations. In Section 4.3, the homothetic transformation of the terminal invariant set is detailed. This homothetic transformation is proposed as a suitable time-varying invariant set to be used in the time-varying MPC formulation. In Section 4.4, the MPC strategy that involves the homothetic transformation, performed on-line, is presented. The time-varying invariant set, result of the homothetic transformation used within the MPC, guarantees the asymptotic stability of the control law. In Section 4.5, a numerical example is presented. Finally, in Section 4.6, the MPC tracking strategy is applied to the *Vir'volt* vehicle.

4.2 Preliminaries

Consider the MPC-based tracking problem (3.11) for the system (3.3) given by

$$\Delta x(k+1) = \mathbf{A}\Delta x(k) + \mathbf{B}\Delta u(k), \tag{4.1}$$

with tracking polytopic time-varying constraints in the input and/or the state, verifying $\forall k \geq 0$

$$\begin{aligned} \Delta x(k) &\in X_\Delta(k) \subseteq \mathbb{R}^n, \\ \Delta u(k) &\in U_\Delta(k) \subseteq \mathbb{R}^p. \end{aligned} \tag{4.2}$$

The notation (k) reflects that the control input and the actual state are constrained within sets of which size and shape might be changing in time. The constraints $X_\Delta(k)$ and $U_\Delta(k)$ follow Remark 1 given in Chapter 3.

The objective is to steer to the origin the tracking error (4.1), while fulfilling the polytopic constraints (4.2) with a MPC-based strategy. The open-loop

problem (3.11) turns into

$$\min_{\Delta u(k+i)\in\mathbb{R}^p,\ i=[0,k+N_p-1]} \sum_{i=0}^{N_p-1} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i)\Big) + \Delta x^T(k+N_p)\mathbf{P}\Delta x(k+N_p),$$

$$\begin{aligned} s.t.\quad & \Delta x(k+i+1) = \mathbf{A}\Delta x(k+i) + \mathbf{B}\Delta u(k+i), && \forall i = 0,\dots,N_p-1,\\ & \Delta x(k+i) \in X_\Delta(k), && \forall i = 0,\dots,N_p-1,\\ & \Delta u(k+i) \in U_\Delta(k), && \forall i = 0,\dots,N_p-1,\\ & \Delta x(k+N_p) \in X_f(k), \end{aligned} \tag{4.3}$$

with N_p, $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{P}$ defined as in Section 3.2.

To ensure stability, convergence of the tracking error to zero, and recursive constraints satisfaction altogether, it is convenient to define the time-varying terminal (closed) set $X_f(k)$ of feasible final states as a parameter-dependent invariant set for the system (3.12) under time-varying constraints (4.2). Having in mind the real-time efficiency as a main priority, the point is that simplicity of implementation must be preserved. And yet, as shown in Section 3.3, obtaining an invariant set is computationally demanding. Notice that in particular, the complexity-index of the polytopic constraints may also vary in time, which means that the shape of the polytopic constraints may vary in time. A method to compute $X_f(k)$ is proposed in this chapter and benefits from an ease of real-time implementation while preserving the stability and convergence properties.

Similarly to Section 3.3, consider the closed-loop system

$$\Delta x(k+1) = (\mathbf{A} + \mathbf{B}K)\Delta x(k), \tag{4.4}$$

obtained from (4.1) using the stabilizing feedback control law $\Delta u(k) = K\Delta x(k)$. As in Section 3.3, the gain $K \in \mathbb{R}^{p\times n}$ is derived from $\mathbf{P}$, which is the solution of the Riccati equation with matrices $\mathbf{R}$ and $\mathbf{Q}$ involved in (4.3).

Following (3.13) for the time-invariant case, the input and the state must satisfy $\Delta u(k) = K\Delta x(k) \in U_\Delta(k)$ with $\Delta x(k) \in X_\Delta(k)$. This is equivalent to

$\Delta x(k) \in \mathcal{X}_\Delta(k)$ where

$$\begin{aligned}\mathcal{X}_\Delta(k) &= \{\Delta x(k) \in \mathbb{R}^n : \Delta x(k) \in X_\Delta(k),\ K\Delta x(k) \in U_\Delta(k)\} \\ &= X_\Delta(k) \cap X_{\Delta u}(U_\Delta(k)),\end{aligned} \tag{4.5}$$

and $X_{\Delta u}(U_\Delta(k))$, as in (3.15), is given by

$$X_{\Delta u}(U_\Delta(k)) = \{\Delta x(k) \in \mathbb{R}^n : K\Delta x(k) \in U_\Delta(k)\}. \tag{4.6}$$

The following assumptions will be considered in the sequel.

Assumption 1 *The steady-state targets $\bar{x}(k)$ and $\bar{u}(k)$ are assumed to be available on-line. Furthermore, the targets $\bar{x}(k)$ and $\bar{u}(k)$ are considered to remain constant within the prediction horizon N_p, i.e. $\bar{x}(k+i) = \bar{x}(k)$ and $\bar{u}(k+i) = \bar{u}(k)$ for $i = 0, ..., N_p$.*

Remark 2 *The assumptions made on the available knowledge of the future constraints can affect substantially the feasibility of the solution. In general, the available information about the constraint can be classified as follows*

- *The constraints are measurable at instant k but its future behaviour is not necessarily known.*
- *Only the boundaries of the constraints are known, but its particular behaviour is unknown.*
- *The present and future behaviour of the constraints are fully known.*

This work does not focus on the problem of ensuring recursive feasibility. To our opinion, this problem is not solvable for the general case with partial information of the future behaviour of the constraints. Thereby, we pose the following assumption (Assumption 2) on the on-line available knowledge. The results could be easily adapted to the cases of different assumptions on the knowledge of future constraints.

Assumption 2 *The time-varying constraints sets $X_\Delta(k+i)$ and $U_\Delta(k+i)$ are available on-line at least for $i = 0, ..., N_p$.*

Assumption 3 *The time-varying constraints sets $X_\Delta(k)$ and $U_\Delta(k)$ are assumed to be convex and compact polytopes that contain the origin in their interior, for all k. $X_\Delta(k)$ and $U_\Delta(k)$ are represented by hyperplane representations described by $\boldsymbol{H}_{X_\Delta}(k)$ and $\boldsymbol{H}_{U_\Delta}(k)$, respectively.*

As a result, similarly to the time-invariant constraints case, the polytopic constraints $\mathcal{X}_\Delta(k)$ are represented by the following hyperplane representation

$$\mathcal{X}_\Delta(k) = \{\Delta x(k) \in \mathbb{R}^n : \mathbf{H}_{\mathcal{X}_\Delta}(k)\Delta x(k) \leq \bar{1}\}, \tag{4.7}$$

where $\mathbf{H}_{\mathcal{X}_\Delta}(k) \in \mathbb{R}^{p_\mathcal{X}(k)\times n}$, $\bar{1} \in \mathbb{R}^{p_\mathcal{X}(k)}$, $p_\mathcal{X}(k)$ is the complexity-index of the polytopic constraints $\mathcal{X}_\Delta(k)$ at time k, and

$$\mathbf{H}_{\mathcal{X}_\Delta}(k) = \begin{bmatrix} \mathbf{H}_{X_\Delta}(k) \\ \mathbf{H}_{U_\Delta}(k)K \end{bmatrix}. \tag{4.8}$$

Assumption 4 *The constraints $\mathcal{X}_\Delta(k)$ follow the trade-off described in Remark 1 of Section 3.2.*

Notice that unlike the time-invariant case presented in Section 3.2, since the constraints (4.8) vary in time, then the invariant terminal set must be recomputed as the constraints change. The invariant sets can be recomputed on-line to solve the MPC problem (4.3), or computed off-line and saved in memory, which may require high memory capacity. It must be stressed that making such sets to be time-varying can be a solution to cope, for example, with feasibility purposes. Indeed, the larger the sets, the greater the feasibility region. Nevertheless, the on-line recomputing of the invariant set is often restrictive if hard real-time constraints such as computation time and memory capacity must be faced. The aim of next section is to propose a solution to handle such a problem.

4.3 Homothetic transformation of the invariant set

As a clue to tackle the aforementioned problem, suppose that a set $\hat{\Omega} \subseteq \mathbb{R}^n$ is given off-line. This set is defined as an invariant set for the system (4.4) and admits a polytopic description. The main idea consists in defining an on-line homothetic transformation, centred in the origin, to obtain an homothetic copy of the invariant set $\hat{\Omega}$ at each time k. This homothetic transformation is characterized by a factor $\alpha(k) \in \mathbb{R}$ such as the resulting convex invariant set $X_f(k) = \alpha(k)\hat{\Omega} \subseteq \mathbb{R}^n$ is an invariant set for the system (4.4) under convex polytopic constraints $\mathcal{X}_\Delta(k)$. It is important to emphasize that since $\mathcal{X}_\Delta(k)$ and $\hat{\Omega}$ are convex sets, then the homothetic transformation always exists.

Remark 3 *The stability does not depend on the feature of the invariant set $\hat{\Omega}$. It holds for any invariant set for the system (4.4). The feature can be chosen to meet specific characteristics all along the homothetic transformations. In practice, the choice can be made according to some heuristics specifying whether the invariant set should be centred or not around the origin, symmetric or not, admitting a prescribed number of vertices, etc. The invariant set $\hat{\Omega}$ can be designed from nominal constraints $\hat{X}_\Delta$ and $\hat{U}_\Delta$ conveniently chosen according to those specifications.*

4.3.1 Principle of the homothetic transformation

Consider a given polytopic invariant set $\hat{\Omega}$. This invariant set is described by the following vertex representation (as in (3.18))

$$\hat{\Omega} = \left\{ \Delta x(k) \in \mathbb{R}^n : \Delta x(k) = \mathbf{V}_{\hat{\Omega}}\mathbf{w},\ \sum_{i=1}^{p_{\hat{\Omega}}} \mathbf{w}(i) = 1,\ \mathbf{w}(i) \geq 0,\ \forall i \right\}, \tag{4.9}$$

where $\mathbf{V}_{\hat{\Omega}} \in \mathbb{R}^{n \times p_{\hat{\Omega}}}$ is the vertices array

$$\mathbf{V}_{\hat{\Omega}} = [v_1\ v_2\ ...\ v_{p_{\hat{\Omega}}}], \tag{4.10}$$

and each column $v_j \in \mathbb{R}^n, j = 1, ..., p_{\hat{\Omega}}$, in $\mathbf{V}_{\hat{\Omega}}$ is the *j-th* vertex of the polytope $\hat{\Omega}$. Besides, $\mathbf{w} \in \mathbb{R}^{p_{\hat{\Omega}}}$, and $p_{\hat{\Omega}}$ is the complexity-index (number of vertices in the

polytope) of the invariant set $\hat{\Omega}$.

On the other hand, consider the polytopic closed-loop constraints $\mathcal{X}_\Delta(k)$ in (4.7). Define $\mathcal{P}_i(k)$ as the *i-th* hyperplane given by the *i-th* facet of the polytope $\mathcal{X}_\Delta(k)$. The hyperplane $\mathcal{P}_i(k)$ verifies

$$\mathcal{P}_i(k) = \{\Delta x(k) \in \mathbb{R}^n : H_i(k)\Delta x(k) = 1\}, \tag{4.11}$$

where $H_i(k) \in \mathbb{R}^{1\times n}$ is the *i-th* row in $\mathbf{H}_{\mathcal{X}_\Delta}(k)$, with $i = 1, ..., p_{\mathcal{X}}(k)$. Notice that $H_i(k)\Delta x(k) = 1$ describes the hyperplane equation of the *i-th* facet of the polytope $\mathcal{X}_\Delta(k)$.

Now, given a vertex $v_j \in \mathbb{R}^n$ $(j = 1, \cdots, p_{\hat{\Omega}})$ in (4.10), such that the scalar $H_i(k)v_j$ is non-zero, there exists a scalar $\beta \in \mathbb{R}$ such that βv_j belongs to the hyperplane $\mathcal{P}_i(k)$ (see Fig. 4.1). Since, this non-zero scalar β belongs to the hyperplane $\mathcal{P}_i(k)$, it verifies

$$H_i(k)\beta v_j = 1. \tag{4.12}$$

It is illustrated in Fig. 4.1. Hereafter, the scalar β will be indexed by i, j and k as follows

$$\beta_{(i,j,k)} = \frac{1}{H_i(k)v_j} \ . \tag{4.13}$$

4.3.2 Computation of the homothetic factor

For the polytopic sets $\hat{\Omega}$ and $\mathcal{X}_\Delta(k)$, define the following strategy from (4.13)

$$\alpha(k) = \min\{\max\{0, \beta_{(i,j,k)}\},\ i = 1, ..., p_{\mathcal{X}}(k),\ j = 1, ..., p_{\hat{\Omega}}\}. \tag{4.14}$$

Such a strategy delivers at each time k a contracting or a dilating factor $\alpha(k)$ such as the set $X_f(k) = \alpha(k)\hat{\Omega} \subseteq \mathcal{X}_\Delta(k)$ is the largest "copy" of $\hat{\Omega}$ contained in $\mathcal{X}_\Delta(k)$. In Fig. 4.1, it can be observed that the homothetic copy of $\hat{\Omega}$ dilates or contracts homogeneously, and its vertices follow the direction of the vertices of $\hat{\Omega}$. The following remark guarantees the existence of such a factor.

Remark 4 *At each time k, since $\mathcal{X}_\Delta(k)$ is assumed to be convex and compact, then for each v_j, $j = 1, ..., p_{\hat{\Omega}}(k)$, there exists at least one $H_i(k)$, $i = 1, ..., p_{\mathcal{X}}(k)$,*

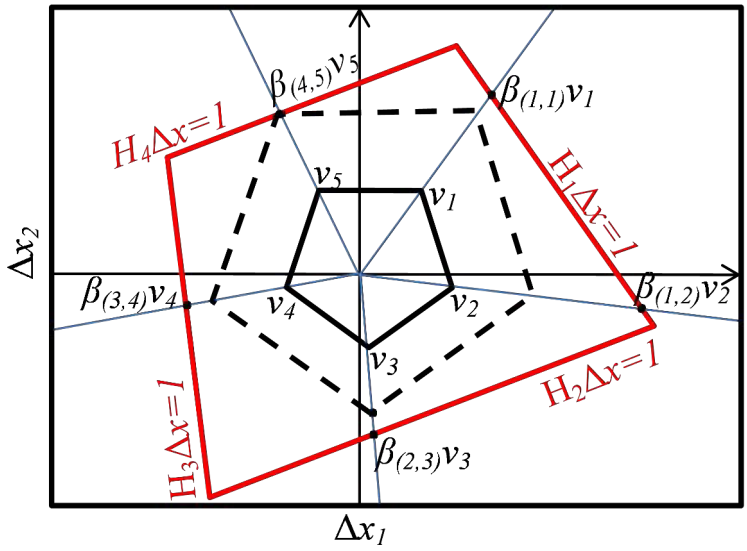


Figure 4.1: Geometry of the homothetic transformation. Solid line: polytopic invariant set $\hat{\Omega}$. Dashed line: homothetic copy of $\hat{\Omega}$ at time k. Red line: polytopic constraints $\mathcal{X}_\Delta$ at time k. $\beta_{(i,j)}v_j$: homothetic vector of v_j (along the blue vectors) with factor $\beta_{(i,j)}$ at time k (time index k not reported for brevity sake).

such that $\beta_{(i,j,k)}$ *is positive.*

The strategy of the homothetic transformation can be summarized as follows

- If $0 < \alpha(k) < 1$, then the invariant set $\hat{\Omega}$ is contracted by a factor α such that $\alpha(k)\hat{\Omega}$ is the largest homothetic contraction of $\hat{\Omega}$ that is contained in $\mathcal{X}_\Delta(k)$.
- If $\alpha(k) = 1$, then the invariant set $\hat{\Omega}$ is already the largest set that is contained in $\mathcal{X}_\Delta(k)$, thus $\hat{\Omega}$ is kept unchanged.
- If $\alpha(k) > 1$, then the invariant set $\hat{\Omega}$ is dilated by a factor α such that $\alpha(k)\hat{\Omega}$ is the largest homothetic dilation of $\hat{\Omega}$ that is contained in $\mathcal{X}_\Delta(k)$.

Notice that nothing has been said yet regarding the invariance properties of the polytopic set $\alpha(k)\hat{\Omega}$. Until now, only a maximal homothetic transformation that fits the time-varying constraints $\mathcal{X}(k)$ has been found from a single invariant set $\hat{\Omega}$. In next section, the homothetic transformation $\alpha(k)\hat{\Omega}$ is proposed as a candidate to be an invariant set for the time-varying problem (4.3), and the MPC problem (4.3) is reformulated using $\alpha(k)\hat{\Omega}$.

4.4 MPC with homothetic transformation of the invariant set

In the following, is demonstrated that the set $\alpha(k)\hat{\Omega}$ is an invariant set for the problem (4.4) with time-varying constraints $X_\Delta(k)$ and $U_\Delta(k)$, and therefore can be used as a terminal set constraint in (4.3).

Proposition 1 *The convex polytope $X_f(k) = \alpha(k)\hat{\Omega}$, under the strategy (4.14), is an invariant set for the system (4.4) under time-varying constraints $\mathcal{X}_\Delta(k)$.*

Proof 1 *Since $\hat{\Omega}$ is an invariant set for (4.4), then $\alpha(k)\hat{\Omega}$ is also an invariant set for (4.4) in absence of constraints (or for no particular constraints), in virtue of the invariance properties for linear systems (see [17]). In addition, since $\alpha(k)\hat{\Omega} \subseteq \mathcal{X}_\Delta(k)$, then $\alpha(k)\hat{\Omega}$ is also an invariant set for (4.4) under constraints $\mathcal{X}_\Delta(k)$ in virtue of Definition 1.*

Therefore, the problem (4.3) can rewritten as follows using $\alpha(k)\hat{\Omega}$

$$\min_{\Delta u(k+i)\in\mathbb{R}^p,\ i=[0,k+N_p-1]} \sum_{i=0}^{N_p-1} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i)\Big) + \Delta x^T(k+N_p)\mathbf{P}\Delta x(k+N_p),$$

$$\begin{aligned} s.t.\quad & \Delta x(k+i+1) = \mathbf{A}\Delta x(k+i) + \mathbf{B}\Delta u(k+i), && \forall i = 0,\dots,N_p-1,\\ & \Delta x(k+i) \in X_\Delta(k), && \forall i = 0,\dots,N_p-1,\\ & \Delta u(k+i) \in U_\Delta(k), && \forall i = 0,\dots,N_p-1,\\ & \Delta x(k+N_p) \in \alpha(k)\hat{\Omega}. \end{aligned} \tag{4.15}$$

Regarding the stability properties of problem (4.15), the following result is obtained as a corollary of Proposition 1.

Corollary 1 *The control law given by the solution of (4.15) with $X_f(k) = \alpha(k)\hat{\Omega}$ and $\alpha(k)$ defined as in (4.14), guarantees the convergence and the stability of $\Delta x(k)$ around the origin.*

Proof 2 *It is a straightforward consequence of Proposition 1 which ensures that $X_f(k) = \alpha(k)\hat{\Omega}$ is an invariant set for (4.4) under constraints $\mathcal{X}_\Delta(k)$. Indeed,*

such a property is sufficient to guarantee the asymptotic stability of $\Delta x(k)$ *around the origin.*

The algorithm used to implement the MPC problem (4.15), can be summarized as follows.

Algorithm 2

- Off-line steps
  Compute an invariant set $\hat{\Omega}$ for the system (4.1), with some arbitrary constraints $\hat{X}_\Delta$ and $\hat{U}_\Delta$ according to Remark 3.
- On-line steps
  1. Measure $\Delta x(k)$ and $\Delta u(k)$ at time k.
  2. Measure $X_\Delta(k)$ and $X_\Delta(k)$ at time k and find $\mathcal{X}_\Delta(k)$ from (4.7) and (4.8).
  3. Compute $\alpha(k)$ using the strategy (4.14) ($\alpha(k)$ verifies $\alpha(k)\hat{\Omega} \subseteq \mathcal{X}_\Delta(k)$).
  4. Solve the MPC problem (4.15) with terminal set constraint $\alpha(k)\hat{\Omega}$.
  5. Apply $\Delta u(k)$ as the first element of the solution of (4.15).
  6. Return to Step 1.

The Algorithm 2 is illustrated in the next section.

4.5 MPC-based tracking with time-varying constraints: an academic example

4.5.1 Problem statement

Consider the same discrete-time system (3.22) as in Section 3.4 given by

$$\mathbf{A} = \begin{bmatrix} 0.9 & 0.25 \\ -0.25 & 0.9 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0.5 \\ 2 \end{bmatrix}. \tag{4.16}$$

The aim is to calculate a control law in order to track the reference given by

$$\bar{x}(k) = \begin{cases} [0.5; 0]^T & \text{for } k < 100, \\ [0.1; 0.4]^T & \text{for } k \geq 100, \end{cases} \tag{4.17}$$

subject to the polytopic time-varying tracking constraints $X_\Delta(k)$ and $U_\Delta(k)$. The constraints $X_\Delta(k)$ are described by

$$X_\Delta(k) = \{\Delta x(k) \in \mathbb{R}^2 : \mathbf{H}_{X_\Delta}(k)\Delta x(k) \leq \bar{1}\}, \tag{4.18}$$

where

$$\mathbf{H}_{X_\Delta}(k) = \begin{cases} \mathbf{H}_{X_\Delta}^{(1)} & \text{for } k < 30, \\ \mathbf{H}_{X_\Delta}^{(2)} & \text{for } 30 \leq k < 90, \\ \mathbf{H}_{X_\Delta}^{(3)} & \text{for } 90 \leq k < 140, \\ \mathbf{H}_{X_\Delta}^{(4)} & \text{for } k \geq 140, \end{cases} \tag{4.19}$$

with

$$\begin{aligned} \mathbf{H}_{X_\Delta}^{(1)} &= \begin{bmatrix} \frac{1}{0.4} & 0 \\ 0 & \frac{1}{0.4} \\ \frac{-1}{0.4} & 0 \\ 0 & \frac{-1}{0.4} \end{bmatrix} = \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \\ -2.5 & 0 \\ 0 & -2.5 \end{bmatrix}, \\ \mathbf{H}_{X_\Delta}^{(2)} = \mathbf{H}_{X_\Delta}^{(4)} &= \begin{bmatrix} \frac{1}{0.1} & 0 \\ 0 & \frac{1}{0.1} \\ \frac{-1}{0.1} & 0 \\ 0 & \frac{-1}{0.1} \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \\ -10 & 0 \\ 0 & -10 \end{bmatrix}, \\ \mathbf{H}_{X_\Delta}^{(3)} &= \begin{bmatrix} \frac{1}{0.3} & 0 \\ 0 & \frac{1}{0.4} \\ \frac{-1}{0.3} & 0 \\ 0 & \frac{-1}{0.4} \end{bmatrix} = \begin{bmatrix} 3.3 & 0 \\ 0 & 2.5 \\ -3.3 & 0 \\ 0 & -2.5 \end{bmatrix}. \end{aligned} \tag{4.20}$$

Besides, the constraints $U_\Delta(k)$ are described by

$$U_\Delta(k) = \{\Delta u(k) \in \mathbb{R}^1 : \mathbf{H}_{U_\Delta}(k)\Delta u(k) \leq \bar{1}\}, \tag{4.21}$$

where

$$\mathbf{H}_{U_\Delta}(k) = \begin{cases} \mathbf{H}_{U_\Delta}^{(1)} & \text{for } k < 30, \\ \mathbf{H}_{U_\Delta}^{(2)} & \text{for } 30 \leq k < 90, \\ \mathbf{H}_{U_\Delta}^{(3)} & \text{for } 90 \leq k < 140, \\ \mathbf{H}_{U_\Delta}^{(4)} & \text{for } k \geq 140, \end{cases} \tag{4.22}$$

with

$$\begin{gathered} \mathbf{H}_{U_\Delta}^{(1)} = \begin{bmatrix} \frac{1}{0.04} \\ \frac{-1}{0.04} \end{bmatrix} = \begin{bmatrix} 25 \\ -25 \end{bmatrix}, \ \mathbf{H}_{U_\Delta}^{(2)} = \mathbf{H}_{U_\Delta}^{(4)} = \begin{bmatrix} \frac{1}{0.01} \\ \frac{-1}{0.01} \end{bmatrix} = \begin{bmatrix} 100 \\ -100 \end{bmatrix}, \\ \mathbf{H}_{U_\Delta}^{(3)} = \begin{bmatrix} \frac{1}{0.05} \\ \frac{-1}{0.04} \end{bmatrix} = \begin{bmatrix} 20 \\ -25 \end{bmatrix}. \end{gathered} \tag{4.23}$$

The feedback gain $K = [-0.0343 \ \ -0.1478]$ is obtained from $\mathbf{Q} = [1\ 0; 0\ 1]$, $\mathbf{R} = 30$ and $\mathbf{P} = [4.6534\ 0.5613; 0.5613\ 3.0237]$, noticing that it does not depend on the constraints. Therefore, from (4.8), the time-varying closed-loop constraints $\mathcal{X}_\Delta(k)$ are given by the following polytopic representation

$$\mathcal{X}_\Delta(k) = \{\Delta x(k) \in \mathbb{R}^2 : \mathbf{H}_{\mathcal{X}_\Delta}(k)\Delta x(k) \leq \bar{1}\}, \tag{4.24}$$

where

$$\mathbf{H}_{\mathcal{X}_\Delta}(k) = \begin{cases} \mathbf{H}_{\mathcal{X}_\Delta}^{(1)} & \text{for } k < 30, \\ \mathbf{H}_{\mathcal{X}_\Delta}^{(2)} & \text{for } 30 \leq k < 90, \\ \mathbf{H}_{\mathcal{X}_\Delta}^{(3)} & \text{for } 90 \leq k < 140, \\ \mathbf{H}_{\mathcal{X}_\Delta}^{(4)} & \text{for } k \geq 140, \end{cases} \tag{4.25}$$

with

$$\mathbf{H}_{\mathcal{X}_\Delta}^{(1)} = \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \\ -2.5 & 0 \\ 0 & -2.5 \\ -0.86 & -3.69 \\ 0.86 & 3.69 \end{bmatrix}, \ \mathbf{H}_{\mathcal{X}_\Delta}^{(2)} = \mathbf{H}_{\mathcal{X}_\Delta}^{(4)} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \\ -10 & 0 \\ 0 & -10 \\ -3.43 & -14.78 \\ 3.43 & 14.78 \end{bmatrix}, \tag{4.26}$$

and

$$\mathbf{H}_{\mathcal{X}_\Delta}^{(3)} = \begin{bmatrix} 3.33 & 0 \\ 0 & 2.5 \\ -3.33 & 0 \\ 0 & -2.5 \\ -0.69 & -2.96 \\ 0.69 & 2.96 \end{bmatrix}. \tag{4.27}$$

In the following section, Algorithm 2 is applied.

4.5.2 Results

- Off-line step: choice of the invariant set.
  According to Remark 3, the choice of the invariant set $\hat{\Omega}$ does not impact the stability and may be made according to some heuristics related to the specificity of the application. Here, as an arbitrary choice, it is built from the feedback gain K indicated above (obtained by solving the LQR problem with weighting matrices $\mathbf{Q} = [1\ 0; 0\ 1]$, $\mathbf{R} = 30$ and $\mathbf{P} = [4.6534\ 0.5613; 0.5613\ 3.0237]$ of the MPC problem (4.3)) and the same constraints (3.25) as in the time-invariant case. Hence, the resulting invariant $\hat{\Omega}$ is defined by (3.27) and is recalled in Fig. 4.2.

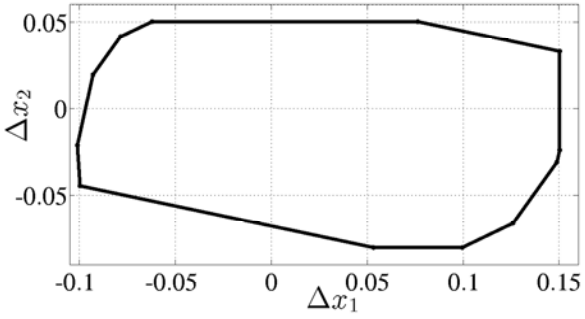


Figure 4.2: Invariant set $\hat{\Omega}$.

- On-line steps: MPC with homothetic transformation.
  By applying the strategy (4.14), the following dilating/contracting homo-

thetic factors are obtained

$$\alpha(k) = \begin{cases} 2.67 & \text{for } k < 30, \\ 0.67 & \text{for } 30 \leq k < 90, \\ 2 & \text{for } 90 \leq k < 140, \\ 0.67 & \text{for } k \geq 140. \end{cases} \tag{4.28}$$

The time evolution of $\alpha(k)$ is plotted in Fig. 4.3. The time-varying constraints (4.18) and (4.21) are depicted in Fig. 4.4, 4.6 and 4.8 for each time interval. The nominal invariant set $\hat{\Omega}$ and the invariant sets obtained after the homothetic transformation with factors (4.28) are depicted in Fig. 4.5, 4.7 and 4.9, respectively. The figures clearly illustrate that for $\alpha(k) > 1$ (*resp.* $\alpha(k) < 1$), the nominal invariant set $\hat{\Omega}$ dilates (*resp.* contracts) homogeneously. From the figures, it is also clear that the largest homothethic invariant set is still a subset of $\mathcal{X}_{\Delta}(k)$, as expected.

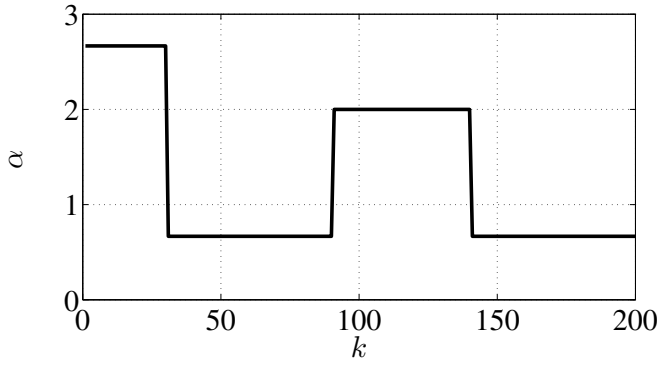


Figure 4.3: Time evolution of $\alpha(k)$.

The dilation/contraction effect of the homothetic factor $\alpha(k)$ over the nominal invariant set $\hat{\Omega}$, is illustrate in the Fig. 4.10.

The tracking of the reference (4.17) with initial conditions $x(0) = [0; 0.3]^T$ is depicted for $x_1(k)$ and $x_2(k)$ in Fig. 4.11 and 4.12, respectively. The tracking errors $\Delta x_1(k)$ and $\Delta x_2(k)$ are depicted in Fig. 4.13 and 4.14. The control $\Delta u(k)$ is the solution of the MPC problem (4.15) with terminal

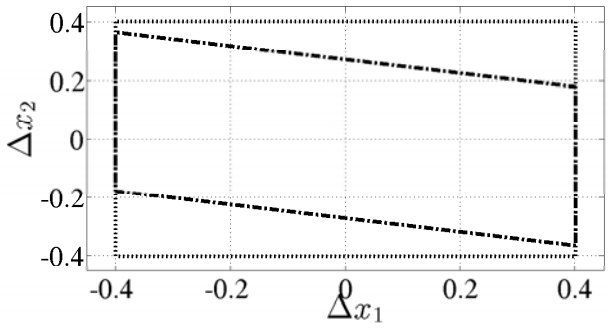


Figure 4.4: Constraints for $k < 30$. Dotted line: constraint $X_\Delta(k)$. Dash-dotted line: constraint $\mathcal{X}_\Delta(k)$.

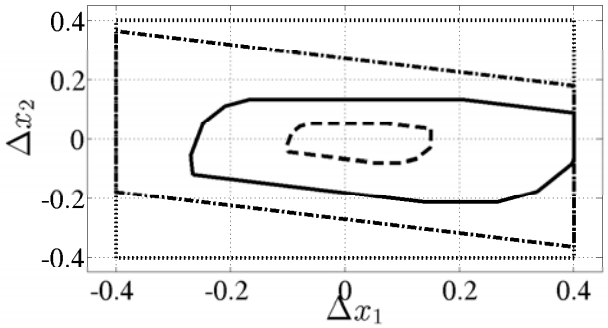


Figure 4.5: Homothetic transformation for $k < 30$. Dotted line: constraint $X_\Delta(k)$. Dash-dotted line: constraint $\mathcal{X}_\Delta(k)$. Dashed line: nominal invariant set $\hat{\Omega}$. Solid line: $\alpha(k)\hat{\Omega}$ with $\alpha(k) = 2.67$.

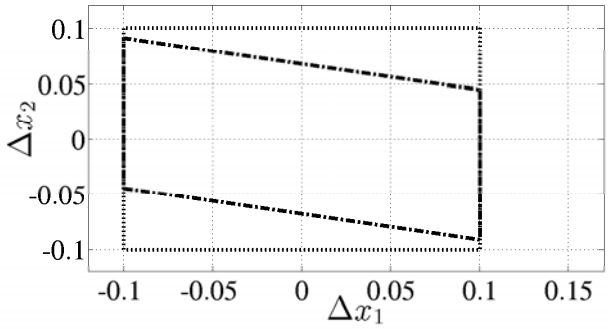


Figure 4.6: Constraints for $30 \leq k < 90$ or $k \geq 140$. Dotted line: constraint $X_\Delta(k)$. Dash-dotted line: constraint $\mathcal{X}_\Delta(k)$.

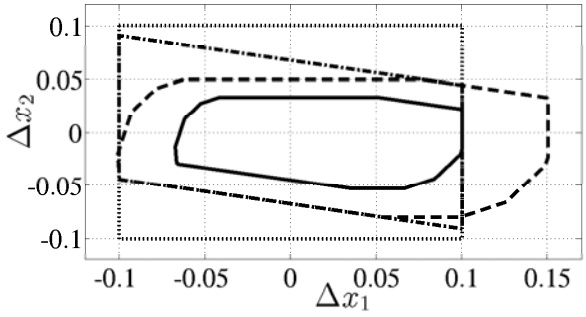


Figure 4.7: Homothetic transformation for $30 \leq k < 90$ or $k \geq 140$. Dotted line: constraint $X_\Delta(k)$. Dash-dotted line: constraint $\mathcal{X}_\Delta(k)$. Dashed line: nominal invariant set $\hat{\Omega}$. Solid line: $\alpha(k)\hat{\Omega}$ with $\alpha(k) = 0.67$.

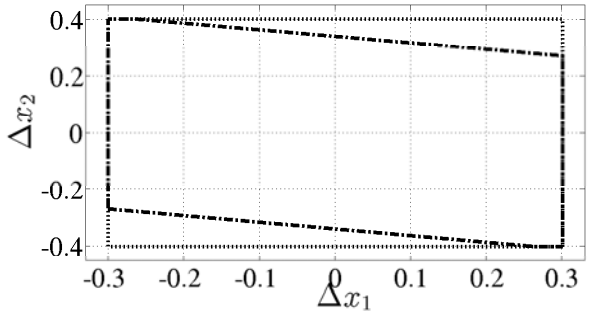


Figure 4.8: Constraints for $90 \leq k < 140$. Dotted line: constraint $X_\Delta(k)$. Dash-dotted line: constraint $\mathcal{X}_\Delta(k)$.

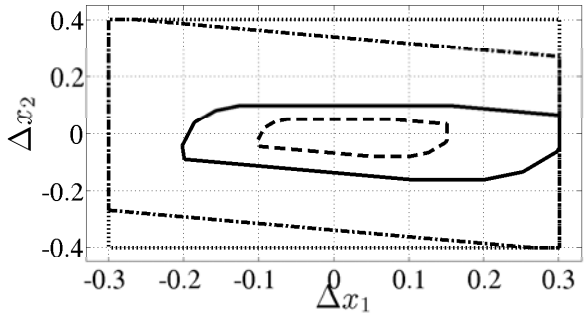


Figure 4.9: Homothetic transformation for $90 \leq k < 140$. Dotted line: constraint $X_\Delta(k)$. Dash-dotted line: constraint $\mathcal{X}_\Delta(k)$. Dashed line: nominal invariant set $\hat{\Omega}$. Solid line $\alpha(k)\hat{\Omega}$ with $\alpha(k) = 2$.

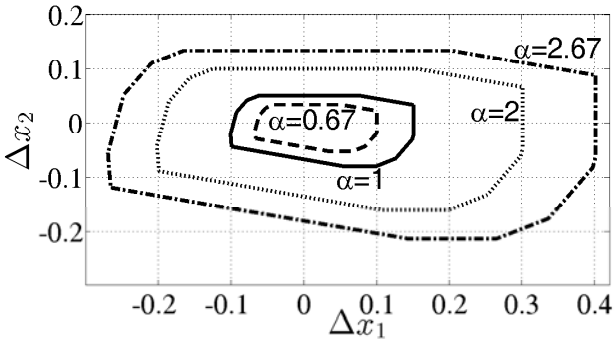


Figure 4.10: Dilation and contraction of the invariant set. Solid line: nominal invariant set $\hat{\Omega}$. Dashed line: $0.67\hat{\Omega}$. Dotted line: $2\hat{\Omega}$. Dash-dotted line: $2.67\hat{\Omega}$.

constraint $\alpha(k)\hat{\Omega}$. The control $\Delta u(k)$ is depicted in Fig. 4.15. The plots highlight that the tracking is achieved while the time-varying constraints are fulfilled.

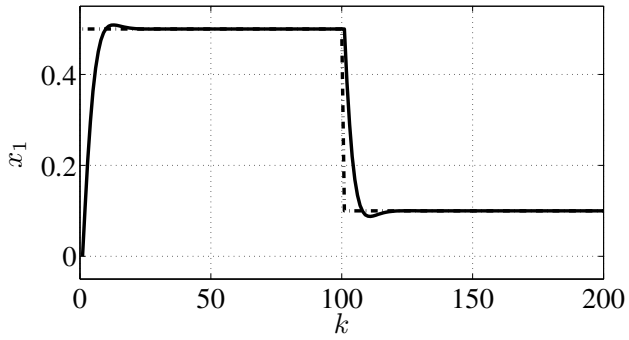


Figure 4.11: Tracking of the reference $\bar{x}_1(k)$. Solid line: tracking response of $x_1(k)$. Dash-dotted line: reference.

4.5.3 Computational resources

The on-line computational requirements (memory and computation time) of the MPC approach with the homothetic transformation are compared with respect to the computational requirements of the standard MPC approach. In the MPC approach with the homothetic transformation, only the dilation/contraction of

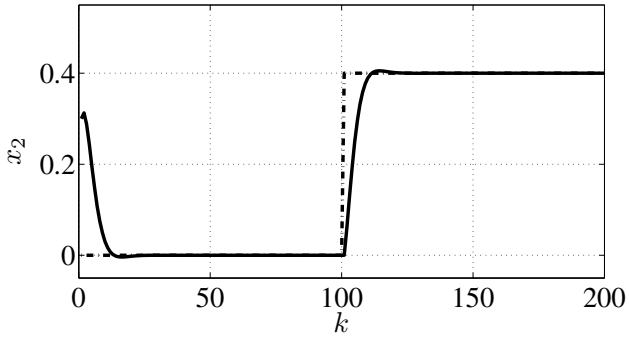


Figure 4.12: Tracking of the reference $\bar{x}_2(k)$. Solid line: tracking response of $x_2(k)$. Dash-dotted line: reference.

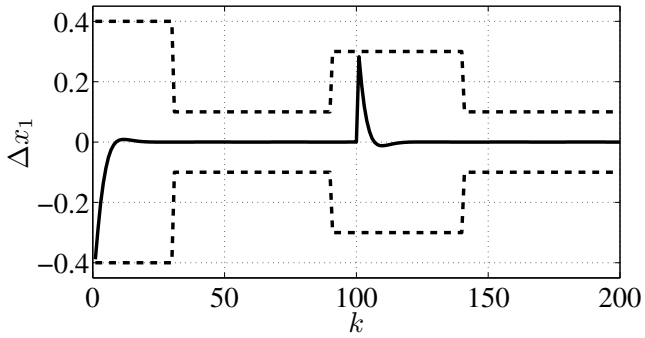


Figure 4.13: Tracking error $\Delta x_1(k)$. Solid line: tracking error $\Delta x_1(k)$. Dashed line: time-varying constraints.

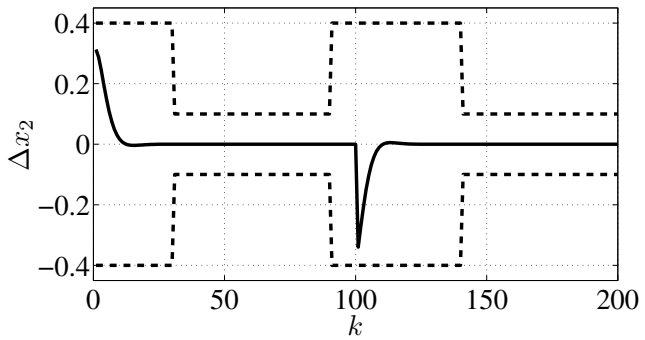


Figure 4.14: Tracking error $\Delta x_2(k)$. Solid line: tracking error and $\Delta x_2(k)$. Dashed line: time-varying constraints.

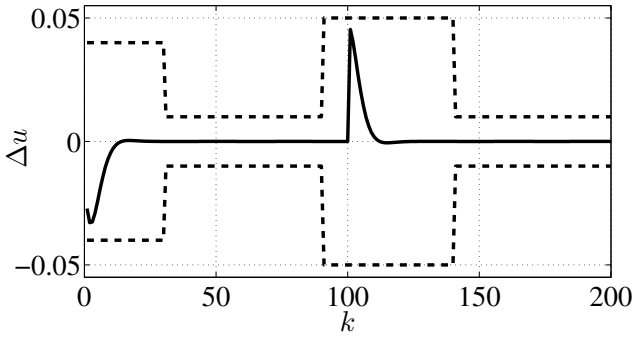


Figure 4.15: Tracking error $\Delta u(k)$. Solid line: tracking error $\Delta u(k)$. Dashed line: time-varying constraints.

the nominal invariant set is performed on-line to fit the time-varying constraints. None invariant set is computed on-line. Conversely, in the standard MPC, a new invariant set is fully recomputed on-line every time the constraints vary in time. In Table 4.1, the computational resources used to solve the tracking of the reference (4.17) with time-varying constraints (4.18) and (4.21) are presented for each approach. The numerical simulations are performed on an Intel Corei7 with 2.2GHz and 6GB of RAM.

Table 4.1: On-line computational resources MPC with homothetic transformation and standard MPC.

Method	**Computation time**	**Memory used**
MPC with homothetic transformation	7.49ms	3002Bytes
Standard MPC	21.28ms	10494Bytes

The MPC approach with the homothetic transformation is almost 3 times faster than the standard MPC approach in which the invariant set is completely recomputed on-line as the constraints vary in time. Additionally, the memory used by all the variables involved in the on-line computation of the homothetic factor $\alpha(k)$, and by those variables involved in solving the MPC problem, is near 3.5 times smaller than the memory used on-line to fully compute the invariant set using the *backward iterative algorithm* and solve the MPC. It illustrates that the homothetic transformation allows to reduce the memory and time requirements for the real-time implementation, in case of time-varying constraints. The

fully recomputation of the invariant set at each time that the constraints vary in time allows to have a bigger domain of attraction, but the performances of both approaches are similar.

4.6 MPC-based tracking with time-varying constraints: application to the Vir'volt vehicle

In this section, the problem (4.3) is fully developed for the linearised model (2.32) of the *Vir'volt* electric vehicle dynamics around (x_{2_e}, I_{batt_e}). Hereafter, the battery current I_{batt} will be simply denoted with u, therefore $I^*_{batt} = u^*$, $I_{batt_e} = u_e$.

The main objective is to solve the MPC law (4.3) to steer to zero the error dynamics (2.33). Now, a bounded set of constraints $X_\Delta(k)$ for every k is imposed. As it turns out, no special constraints are required for the position accuracy $\Delta x_1(k)$ since the tracking is only concerned with the velocity. For that reason, in practice, very large constraints $-100\text{m} \leq \Delta x_1(k) \leq 100\text{m}$ are imposed for every k.

Regarding the tracking constraints on the velocity, it is worth considering robustness issues. Indeed, there are unavoidable mismatches between the parameters of the model and the actual ones. Discrepancies regarding the mass of the vehicle or the frictions will be more critical during the initial acceleration and the final deceleration (see Fig. 2.8a). For example, an increasing of the mass will lead to a slower acceleration. Therefore, the constraints for $\Delta x_2(k)$ must be relaxed at the beginning and at the end of the path. Consequently, the constraints for $\Delta x_2(k)$ are time-varying.

Notice that, since the vehicle starts at speed equal to zero, the initial natural strategy is to accelerate as much as possible to reach the optimal nominal velocity. That precisely corresponds to the optimal solution as illustrated by Fig. 2.8a and Fig. 2.8b. Notice also that when the vehicle approaches the end of the path, the natural energetically optimal behaviour consists in switching off the motor propulsion (see Fig. 2.8a).

The resulting constraints $X_\Delta(k)$ are represented by the following polytopic

description $\mathbf{H}_{X_\Delta}(k)$ as defined in (4.18)

$$\mathbf{H}_{X_\Delta}(k) = \begin{cases} \mathbf{H}_{X_\Delta}^{(1)} & \text{for } x_1^*(k) < 944\text{m}, \\ \mathbf{H}_{X_\Delta}^{(2)} & \text{for } 944\text{m} \leq x_1^*(k) < 2588\text{m}, \\ \mathbf{H}_{X_\Delta}^{(3)} & \text{for } x_1^*(k) \geq 2588\text{m}, \end{cases} \tag{4.29}$$

where

$$\mathbf{H}_{X_\Delta}^{(1)} = \begin{bmatrix} \frac{1}{100} & 0 \\ 0 & \frac{1}{0.28} \\ \frac{-1}{100} & 0 \\ 0 & \frac{-1}{1.67} \end{bmatrix} = \begin{bmatrix} 0.01 & 0 \\ 0 & 3.6 \\ -0.01 & 0 \\ 0 & -0.6 \end{bmatrix}, \mathbf{H}_{X_\Delta}^{(2)} = \begin{bmatrix} \frac{1}{100} & 0 \\ 0 & \frac{1}{0.28} \\ \frac{-1}{100} & 0 \\ 0 & \frac{-1}{0.83} \end{bmatrix} = \begin{bmatrix} 0.01 & 0 \\ 0 & 3.6 \\ -0.01 & 0 \\ 0 & -1.2 \end{bmatrix},$$
$$\mathbf{H}_{X_\Delta}^{(3)} = \begin{bmatrix} \frac{1}{100} & 0 \\ 0 & \frac{1}{1.66} \\ \frac{-1}{100} & 0 \\ 0 & \frac{-1}{0.55} \end{bmatrix} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.6 \\ -0.1 & 0 \\ 0 & -1.8 \end{bmatrix}. \tag{4.30}$$

As far as the input tracking constraints $U_\Delta(k)$ are concerned, the battery current $u(k)$ must fulfil, for all k

$$0 \leq u(k) \leq u_{max}. \tag{4.31}$$

Then, to avoid the inadmissible values of the input, $\Delta u(k)$ in (2.32) has to satisfy the following constraint

$$\begin{aligned} \Delta u_{min}(k) \leq & \Delta u(k) \leq \Delta u_{max}(k), \\ -u^*(k) \leq & \Delta u(k) \leq u_{max} - u^*(k). \end{aligned} \tag{4.32}$$

The constraints are time-varying because $\Delta u(k)$ depends on the optimal control input $u^*(k)$ (see Fig. 2.8b) that may change in time. Additionally, $0 \in int(U_\Delta(k))$ must be satisfied for every k, and therefore a small tolerance $\epsilon = 1 \times 10^{-6}$ is introduced for the cases where $u^*(k) = u_{max}$ or $u^*(k) = 0$. As a result, the

constraints $U_\Delta(k)$ on the input, as defined in (4.21), obey

$$\mathbf{H}_{U_\Delta}(k) = \begin{cases} \mathbf{H}_{U_\Delta}^{(1)} & \text{if } u^*(k) = u_{max}, \\ \mathbf{H}_{U_\Delta}^{(2)} & \text{if } u^*(k) = 0, \\ \mathbf{H}_{U_\Delta}^{(3)} & \text{otherwise,} \end{cases} \tag{4.33}$$

where

$$\mathbf{H}_{U_\Delta}^{(1)} = \begin{bmatrix} (\epsilon)^{-1} \\ (-u_{max})^{-1} \end{bmatrix}, \ \mathbf{H}_{U_\Delta}^{(2)} = \begin{bmatrix} (u_{max})^{-1} \\ (-\epsilon)^{-1} \end{bmatrix}, \ \mathbf{H}_{U_\Delta}^{(3)} = \begin{bmatrix} (u_{max} - u^*(k))^{-1} \\ (-u^*(k))^{-1} \end{bmatrix}. \tag{4.34}$$

The time-varying tracking constraints for $\Delta u(k)$ (see (4.32)) are depicted in Fig. 4.16 with respect to the vehicle position. Notice that the constraints $U_\Delta(k)$ are not symmetric.

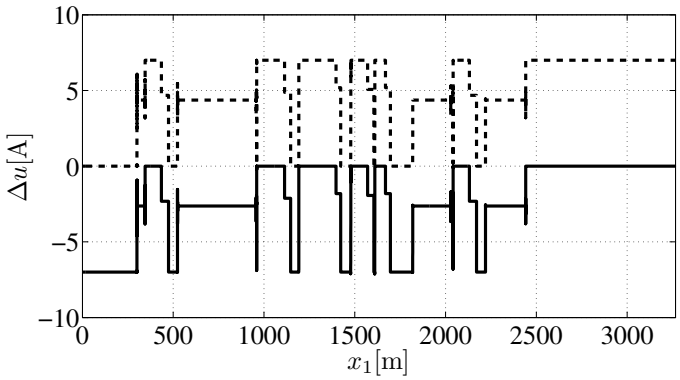


Figure 4.16: Time-varying tracking constraints $U_\Delta(k)$ with respect to the vehicle's position. Solid line: $\Delta u_{min}(k)$. Dashed line: $\Delta u_{max}(k)$.

4.6.1 Results

In order to make the tracking task appropriate for real-time implementation despite the time-varying constraints, Algorithm 2 is applied. To this end, a nominal invariant set $\hat{\Omega}$ must be precomputed off-line for (2.32) and then scaled on-line to fit the time-varying constraints described by (4.29) and (4.33), as seen in Section 4.3. This invariant set will act as a terminal set constraint and stability and

convergence of the state to steady-state values will be guaranteed by the MPC algorithm (see Corollary 1).

- Off-line step: The choice of the invariant set.
  The nominal tracking constraints (used only to compute off-line the initial invariant set) $\hat{X}_\Delta$ and $\hat{U}_\Delta$ (see Remark 3), are chosen as

$$\begin{aligned} -50\text{m} \leq &\Delta x_1(k) \leq 50\text{m}, \\ -0.138\text{m/s} \leq &\Delta x_2(k) \leq 0.138\text{m/s}, \end{aligned} \tag{4.35}$$

  and

$$-10\text{A} \leq \Delta u(k) \leq 10\text{A}. \tag{4.36}$$

  The polytopes $\hat{X}_\Delta$ and $\hat{U}_\Delta$ are described by

$$\mathbf{H}_{\hat{X}_\Delta} = \begin{bmatrix} \frac{1}{50} & 0 \\ 0 & \frac{1}{0.138} \\ \frac{-1}{50} & 0 \\ 0 & \frac{-1}{0.138} \end{bmatrix} = \begin{bmatrix} 0.02 & 0 \\ 0 & 7.20 \\ -0.02 & 0 \\ 0 & -7.20 \end{bmatrix}, \ \mathbf{H}_{\hat{U}_\Delta} = \begin{bmatrix} \frac{1}{10} \\ \frac{-1}{10} \end{bmatrix} = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}. \tag{4.37}$$

  The weighting matrices are $\mathbf{Q} = [0\,0; 0\,1]$ and $\mathbf{R} = 1$, thus $\mathbf{P} = [0\,0; 0\;139.75]$ and $K = [0 \;\; -0.6411]$. The following closed-loop nominal constraints $\hat{\mathfrak{X}}_\Delta$ are obtained with K defined as above,

$$\mathbf{H}_{\hat{\mathfrak{X}}_\Delta} = \begin{bmatrix} 0.02 & 0 \\ 0 & 7.20 \\ -0.02 & 0 \\ 0 & -7.20 \\ 0 & -0.064 \\ 0 & 0.064 \end{bmatrix}. \tag{4.38}$$

The maximal invariant set $\hat{\Omega}$ that fits (4.38) is given by the following vertex representation

$$\mathbf{V}_{\hat{\Omega}} = \begin{bmatrix} 50 & -0.1389 \\ 50 & -0 \\ 49.7020 & 0.1389 \\ -49.7020 & -0.1389 \\ -50 & 0.1389 \\ -50 & 0 \end{bmatrix}^T . \tag{4.39}$$

- On-line steps: MPC with homothetic transformation.
  Since in the actual application, the dynamics of the vehicle is nonlinear, the tracking task will be performed by applying the control to the nonlinear discrete-time dynamics (2.22), as is depicted in the block diagram of Fig. 4.17. The optimal solution $\Delta u(k), \Delta u(k+1), ..., \Delta u(k+N_p)$ of the MPC problem (4.3), with $N_p = 10$, is computed from the actual state $x(k) = [x_1(k), x_2(k)]^T$ of the nonlinear dynamics (2.22) using the linearised tracking error $\Delta x_2(k) = (x_2(k) - x_{2_e}) - (x_2^*(k) - x_{2_e})$. The operating point is fixed as the average velocity of the optimal solution depicted in Fig. 2.8, i.e. $x_{2_e} = 27$km/h.

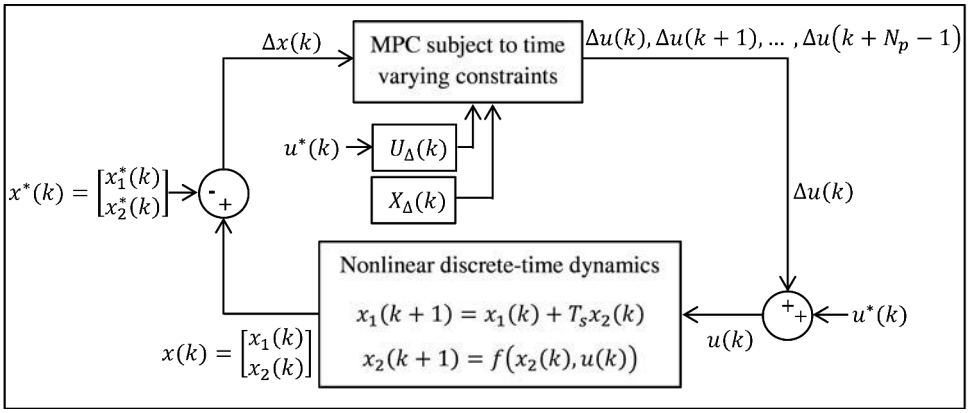


Figure 4.17: Closed-loop implementation of the time-varying MPC. The function that describes the velocity $x_2(k+1) = f(x_2(k), u(k))$ is given by (2.22).

The homothetic dilation/contraction of the invariant set is applied within the MPC algorithm to fit the closed-loop constraints $\mathcal{X}_\Delta(k)$, according to Algorithm 2. The first component $\Delta u(k)$ of the optimal solution of the

MPC problem is used to shape the control input $u(k)$ for the nonlinear dynamics by making $u(k) = \Delta u(k) + u^*(k)$. Notice that to comply with Assumption 1, the references $u^*(k)$ and $x^*(k)$ are kept constant during the prediction horizon N_p. The sampling period is $T_s = 0.2$s. The tracking task is performed for a mass variation of 10% and 50% in the nonlinear model (2.22).

From the strategy (4.14), the scaling factor is calculated on-line and it is obtained that $\alpha(k) \in \{0.067,\ 0.561,\ 1.752,\ 2\}$. Next, the nominal invariant set $\hat{\Omega}$, described by (4.39), is scaled with the homothetic transformation to fit the closed-loop constraints $\mathcal{X}_\Delta(k)$ given by (4.7), where

$$\mathbf{H}_{\mathcal{X}_\Delta}(k) = \begin{bmatrix} \mathbf{H}_{X_\Delta}(k) \\ \mathbf{H}_{U_\Delta}(k)K \end{bmatrix}, \tag{4.40}$$

with $\mathbf{H}_{X_\Delta}(k)$ and $\mathbf{H}_{U_\Delta}(k)$ as given by (4.29) and (4.33), respectively.

The invariant set $\hat{\Omega}$ and the different homothetic transformations $\alpha(k)\hat{\Omega}$ for the distinct values of α are depicted in Fig. 4.18. The evolution of α is plotted in Fig. 4.19 with respect to the vehicle position.

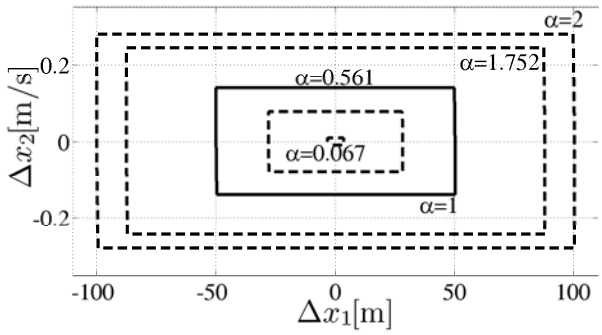


Figure 4.18: Dilation and contraction of the invariant set. Solid line: nominal invariant set $\hat{\Omega}$. Dashed lines: dilation or contraction of the nominal invariant set.

The tracking response $x_2(k)$ of the nonlinear system is depicted in Fig. 4.20 with respect to the vehicle position. The difference between the actual nonlinear velocity $x_2(k)$ and the target state $x^*(k)$ at time k, i.e. $\Delta x_2(k) =$

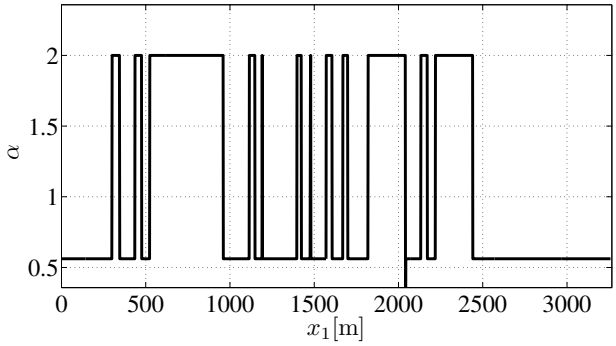


Figure 4.19: Homothetic factor α with respect to the vehicle's position x_1.

$x_2(k) - x_2^*(k)$, is depicted in Fig. 4.21 with respect to the vehicle position. In Fig. 4.22, the input for the nonlinear tracking $\Delta u(k) = u(k) - u^*(k)$ is depicted with respect to the vehicle position. The results show that tracking task is achieved and the constraints are satisfied.

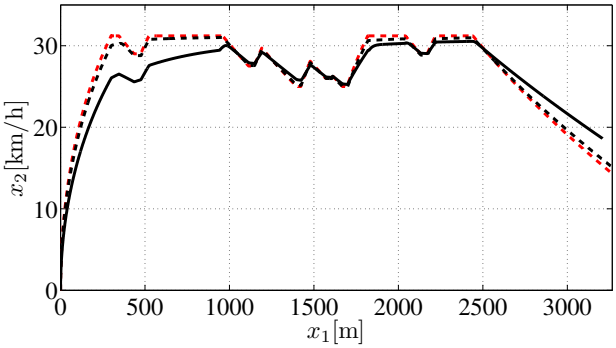


Figure 4.20: Closed-loop tracking response. Red dashed line: reference $x_2^*(k)$. Dashed line: tracking response of the nonlinear system with a mass variation of 10%. Solid line: tracking response of the nonlinear system with a mass variation of 50%.

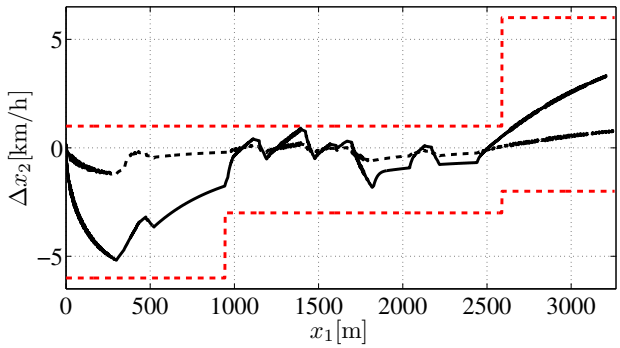


Figure 4.21: Closed-loop tracking error $\Delta x_2(k)$. Red dashed lines: $\Delta x_{2min}(k)$ and $\Delta x_{2max}(k)$. Dashed line: tracking error $\Delta x_2(k)$ of the nonlinear system with a mass variation of 10%. Solid line: tracking error $\Delta x_2(k)$ of the nonlinear system with a mass variation of 50%.

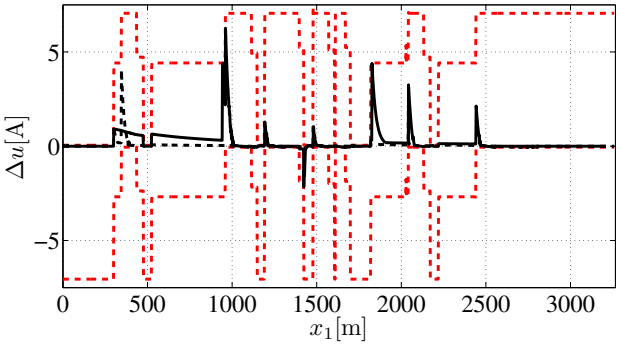


Figure 4.22: Closed-loop tracking error $\Delta u(k)$. Red dashed lines: $\Delta u_{min}(k)$ and $\Delta u_{max}(k)$. Dashed line: tracking input $\Delta u(k)$ of the nonlinear system with a mass variation of 10%. Solid line: tracking input $\Delta u(k)$ of the nonlinear system with a mass variation of 50%.

4.7 Conclusions

A real-time MPC-based tracking strategy for linear systems subject to time-varying constraints in the state and/or the input has been presented. The central idea is based on a polytopic invariant set computed off-line which is homogeneously dilated or contracted on-line to fit polytopic time-varying constraints. The resulting time-varying invariant set has been used as an admissible terminal

constraint set so that it guarantees stability and convergence in the tracking task. It has been shown that time-varying constraints can allow to take into account practical concerns such as the consideration of saturations and the guarantee of feasibility despite uncertainties. The MPC strategy has been applied to the practical case of the *Vir'volt* vehicle.

Beyond the interest of a solution which is appropriate to cope with time-varying constraints, the approach is well suited for real-time applications. Indeed, the additional cost of the approach, compared with the standard MPC for the time-invariant case, is quite negligible.

Until now, the linearised model has been used to design the control law. It should be interesting to account for all the nonlinear features of the tracking problem (due to the nonlinearities of the dynamics of the *Vir'volt* vehicle) in the synthesis of the control law. In the following chapter, the Linear Parametric Representation (LPV) of the tracking problem, detailed in Section 2.5.2, is considered. A MPC-based tracking strategy for LPV systems is developed in order to achieve the tracking task.

Chapter 5

Real-time Robust Model Predictive Control for LPV systems

5.1 Introduction

In this chapter, a control strategy based on the Linear Parametric-Varying (LPV) representation of the tracking error (2.40) in Section 2.5.2 is presented. When it comes to control synthesis, the point which must be considered for LPV representation is that, although the varying parameters are accessible because they are measured on-line, the future behaviour is uncertain. However, if the parameters are bounded, they can be considered as lying in a polytope. In such a case, the control law can result from the solution of a finite number of Linear Matrix Inequalities (LMIs). Furthermore, LMIs are very handy to take into account the uncertainties in the parameters and the constraints imposed to the control task.

The LMIs were first introduced into a robust constrained MPC algorithm for polytopic LTV systems (with a straightforward possible extension to polytopic LPV systems) in [42]. However, the formulation presented in [42] is very conservative. Since then, several works such as [49, 66, 22] have proposed enhancements of [42] to reduce conservativeness. Nevertheless, they require to solve on-line a convex optimization problem involving LMIs, which is a major draw-

back for real-time applications. Furthermore, all the aforementioned works deal with symmetric constraints in the input and the output. On the other hand, the work of [71] addresses the problem of MPC for LPV systems subject to asymmetric constraints. However, it also requires solving on-line LMIs. Besides, as [71] highlights, the proposed solution for asymmetric constraints is very conservative since the control design does not consider the time-varying parameter of the LPV representation. On the other hand, the robust MPC strategy for LPV systems in [66], explicitly includes the measure of the time-varying parameter in the control law, which reduces the conservatism, but it still needs solving on-line of LMIs.

In this chapter, [66] is adapted to obtain an off-line version acceptable for real-time implementation. The modifications partially follow the same lines as in [68] to reduce the computational time and the resources required for the MPC algorithm. However, the off-line algorithm presented in [68] does not take into account the value of the time-varying parameter of the LPV representation. In this chapter, the time variations of the parameter are considered to design a controller suitable for real-time implementation and with low conservatism.

This chapter is organized as follows. In Section 5.2, the problem of MPC tracking for a LPV system is introduced. In Section 5.3, a robust MPC for LPV systems is described. This approach involves the on-line solution of a finite number of LMIs. In Section 5.3.1, an explicit MPC strategy for LPV systems is presented. This approach reduces the on-line computational cost, since the LMIs are solved off-line, but does not include the time-varying parameter in the solution of the control law. In Section 5.5, a robust MPC for LPV systems that includes the time-varying parameter in the computation of the control law is presented. This approach uses a Parameter Dependent Lyapunov Function (PDLF) to reduce conservatism. However, this approach also involves the on-line solution of a finite number of LMIs. In Section 5.6, a new explicit MPC strategy for LPV systems is presented. This strategy involves the time-varying parameter in the control law and reduces the on-line computational cost. Finally, in Sections 5.4 and 5.7, the aforementioned approach is illustrated through numerical examples, and are applied to the problem of the *Vir'volt* vehicle tracking the optimal driving strategy.

5.2 Problem formulation

Consider the dynamics of the polytopic LPV system (2.40), i.e.

$$\Delta x(k+1) = \mathbf{A}(\lambda(k))\Delta x(k) + \mathbf{B}\Delta u(k), \tag{5.1}$$

with $u(k) = I_{batt}(k)$ and $I^*_{batt}(k) = u^*(k)$, and its output equation

$$\Delta y(k) = \mathbf{C}\Delta x(k), \tag{5.2}$$

with

$$\mathbf{A}(\lambda(k)) = \begin{bmatrix} 1 & T_s \\ 0 & 1 - \frac{1}{2m}T_s\rho C_d A_f \lambda(k) \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0 \\ T_s \frac{\eta k_t g_r}{m r_w} \end{bmatrix} \text{ and } \mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \tag{5.3}$$

where $\lambda(k) = (x_2(k) + x^*_2(k))$. The boundaries of the parameter $\lambda(k)$, i.e. $\lambda_{min} \leq \lambda(k) \leq \lambda_{min}$, are assumed to be known but the future behaviour of the parameter is unknown. From Section 2.5.2, it has been already established that the dynamics of the tracking error $\Delta x(k) = [x_1(k) - x^*_1(k); x_2(k) - x^*_2(k)]^T$, belongs to a family of dynamics given by (2.38) within the polytope $\mathcal{C}$. The polytope $\mathcal{C}$ has $p_{\mathcal{C}} = 2$ vertices. In the following, the tracking problem is formulated for the family of dynamics $[\mathbf{A}(\lambda(k))\ \mathbf{B}]$ given by (2.38), i.e.

$$[\mathbf{A}(\lambda(k))\ \mathbf{B}] = \sum_{j=1}^{2} f_j(\lambda(k))[\mathbf{A}_j\ \mathbf{B}], \tag{5.4}$$

with

$$\mathbf{A}_1 = \begin{bmatrix} 1 & T_s \\ 0 & 1 - \frac{1}{2m}T_s\rho C_d A_f \lambda_{min} \end{bmatrix}, \ \mathbf{A}_2 = \begin{bmatrix} 1 & T_s \\ 0 & 1 - \frac{1}{2m}T_s\rho C_d A_f \lambda_{max} \end{bmatrix} \tag{5.5}$$

and

$$f_1(\lambda(k)) = \frac{\lambda_{max} - \lambda(k)}{\lambda_{max} - \lambda_{min}}, \ f_2(\lambda(k)) = \frac{\lambda(k) - \lambda_{min}}{\lambda_{max} - \lambda_{min}}. \tag{5.6}$$

The dynamics (5.1) is subject to polytopic symmetric constraints in the input and in the output, as follows

$$\begin{aligned} \Delta u(k) \in U_\Delta, \ \forall k \geq 0, \\ \Delta y(k) \in Y_\Delta, \ \forall k \geq 0, \end{aligned} \tag{5.7}$$

where U_Δ and Y_Δ are described by

$$U_\Delta = \{\Delta u(k) \in \mathbb{R} : \mathbf{H}_{U_\Delta} \Delta u(k) \leq \bar{1}\} \text{ and } Y_\Delta = \{\Delta y(k) \in \mathbb{R} : \mathbf{H}_{Y_\Delta} \Delta y(k) \leq \bar{1}\}, \tag{5.8}$$

with

$$\mathbf{H}_{U_\Delta} = \begin{bmatrix} \frac{1}{\Delta u_{max}} \\ -\frac{1}{\Delta u_{max}} \end{bmatrix} \text{ and } \mathbf{H}_{Y_\Delta} = \begin{bmatrix} \frac{1}{\Delta y_{max}} \\ -\frac{1}{\Delta y_{max}} \end{bmatrix}. \tag{5.9}$$

As a result, the constraints (5.7) are inequalities constraints of the form

$$\begin{aligned} -\Delta u_{max} \leq \Delta u(k) \leq \Delta u_{max}, \ \forall k \geq 0, \\ -\Delta y_{max} \leq \Delta y(k) \leq -\Delta y_{max}, \ \forall k \geq 0. \end{aligned} \tag{5.10}$$

The MPC tracking, which consists in steering the error $\Delta x(k)$, in (5.1), to zero, can be addressed by the following *min-max* problem, written in terms of the family of dynamics $[\mathbf{A}(\lambda(k)) \ \mathbf{B}]$ (see [42, 68])

$$\min_{F(k+i) \in \mathbb{R}^{p \times n}} \left(\max_{[\mathbf{A}(\lambda(k+i)) \ \mathbf{B}] \in \mathcal{C}, i \geq 0} J(k) \right),$$

$$\begin{aligned} &s.t. \\ &\Delta x(k+i+1) = \mathbf{A}(\lambda(k+i))\Delta x(k+i) + \mathbf{B}\Delta u(k+i), && \forall i \geq 0, \\ &\Delta u(k+i) = F(k+i)\Delta x(k+i), && \forall i \geq 0, \\ &\Delta y(k+i) = \mathbf{C}\Delta x(k+i), && \forall i \geq 1, \\ &\Delta u(k+i) \in U_\Delta, && \forall i \geq 0, \\ &\Delta y(k+i) \in Y_\Delta, && \forall i \geq 1, \end{aligned} \tag{5.11}$$

where the robust performance objective $J(k)$ is given by

$$J(k) = \sum_{i=0}^{\infty} \Big(\Delta x^T(k+i)\mathbf{Q}\Delta x(k+i) + \Delta u^T(k+i)\mathbf{R}\Delta u(k+i) \Big). \tag{5.12}$$

The symmetric weighting matrices $\mathbf{Q} \in \mathbb{R}^{n\times n}, \mathbf{Q} > 0$, and $\mathbf{R} \in \mathbb{R}^{p\times p}, \mathbf{R} > 0$, define the state and the input tracking costs respectively. U_Δ and Y_Δ are the sets of symmetric tracking constraints for the tracking of the input and the output, respectively. As a result, the state feedback control law $\Delta u(k) = F(k)\Delta x(k)$, is the control which minimizes the worst-case of the cost $J(k)$ in (5.11) [42, 68].

5.3 Robust constrained MPC for LPV systems

According to [42, 68], the problem (5.11) is not computationally tractable. However, it can be reformulated as

$$\min_{F(k+i) \in \mathbb{R}^{p\times n}} \gamma(k),$$

$$\begin{aligned}
&s.t.\\
&\Delta x(k+i+1) = \mathbf{A}(\lambda(k+i))\Delta x(k+i) + \mathbf{B}\Delta u(k+i), && \forall i \geq 0,\\
&\Delta u(k+i) = F(k+i)\Delta x(k+i), && \forall i \geq 0,\\
&\Delta y(k+i) = \mathbf{C}\Delta x(k+i), && \forall i \geq 1,\\
&\Delta u(k+i) \in U_\Delta, && \forall i \geq 0,\\
&\Delta y(k+i) \in Y_\Delta, && \forall i \geq 1,
\end{aligned} \tag{5.13}$$

where the scalar $\gamma(k)$ is an upper bound of $J(k)$, in (5.12), at time k [42, 68].

It is shown in [42, 68] that, if there exists a Lyapunov function $V(\Delta x(k))$ to the problem (5.13), such that

$$\max_{[\mathbf{A}(\lambda(k+i))\ \ \mathbf{B}] \in \mathcal{C},\, i\geq 0} J(k) \leq V(\Delta x(k)) \leq \gamma(k), \tag{5.14}$$

for any $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$, $i \geq 0$, with

$$\Delta u(k) = F(k)\Delta x(k), \tag{5.15}$$

then, the existence of the upper bound $\gamma(k)$ is guaranteed, and the problem (5.13), and thus (5.11), is solved by

$$\min_{\boldsymbol{\Theta}(k),\ \boldsymbol{\Lambda}(k)} \gamma(k), \tag{5.16}$$

$s.t.$

$$\begin{bmatrix} 1 & \star \\ \Delta x(k) & \boldsymbol{\Theta}(k) \end{bmatrix} \geq 0,\ \boldsymbol{\Theta}(k) > 0, \tag{5.17}$$

$$\begin{bmatrix} \boldsymbol{\Theta}(k) & \star & \star & \star \\ \mathbf{A}_1\boldsymbol{\Theta}(k) + \mathbf{B}\boldsymbol{\Lambda}(k) & \boldsymbol{\Theta}(k) & \star & \star \\ \mathbf{Q}^{1/2}\boldsymbol{\Theta}(k) & 0 & \gamma(k)I & \star \\ \mathbf{R}^{1/2}\boldsymbol{\Lambda}(k) & 0 & 0 & \gamma(k)I \end{bmatrix} \geq 0, \tag{5.18}$$

$$\begin{bmatrix} \boldsymbol{\Theta}(k) & \star & \star & \star \\ \mathbf{A}_2\boldsymbol{\Theta}(k) + \mathbf{B}\boldsymbol{\Lambda}(k) & \boldsymbol{\Theta}(k) & \star & \star \\ \mathbf{Q}^{1/2}\boldsymbol{\Theta}(k) & 0 & \gamma(k)I & \star \\ \mathbf{R}^{1/2}\boldsymbol{\Lambda}(k) & 0 & 0 & \gamma(k)I \end{bmatrix} \geq 0, \tag{5.19}$$

$$\begin{bmatrix} \Delta u_{max}^2 \boldsymbol{I} & \star \\ \boldsymbol{\Lambda}(k)^T & \boldsymbol{\Theta}(k) \end{bmatrix} \geq 0, \tag{5.20}$$

$$\begin{bmatrix} \Delta y_{max}^2 \boldsymbol{I} & \star \\ (\mathbf{A}_1\boldsymbol{\Theta}(k) + \mathbf{B}\boldsymbol{\Lambda}(k))^T \mathbf{C}^T & \boldsymbol{\Theta}(k) \end{bmatrix} \geq 0, \tag{5.21}$$

$$\begin{bmatrix} \Delta y_{max}^2 \boldsymbol{I} & \star \\ \left(\mathbf{A}_2\boldsymbol{\Theta}(k) + \mathbf{B}\boldsymbol{\Lambda}(k)\right)^T \mathbf{C}^T & \boldsymbol{\Theta}(k) \end{bmatrix} \geq 0, \tag{5.22}$$

($\star$ denotes the corresponding block to make each matrix inequality symmetric) where $\mathbf{A}_1, \mathbf{A}_2$, $\mathbf{B}$ and $\mathbf{C}$ are given in (5.3) and (5.5). The matrix $\boldsymbol{I}$ is the identity matrix. The unknown quantities are $\boldsymbol{\Theta}(k) \in \mathbb{R}^{n\times n}$ and $\boldsymbol{\Lambda}(k) \in \mathbb{R}^{p\times n}$. The quantities $\gamma(k)$, $\boldsymbol{\Theta}(k)$, $\boldsymbol{\Lambda}(k)$, are dependent on time k because they are computed at each execution time k.

To sum up, the solution to the problem (5.16) is the minimum $\gamma(k)$ such that there exist matrices $\boldsymbol{\Theta}(k)$ and $\boldsymbol{\Lambda}(k)$ that satisfy the LMIs stated in (5.17)-(5.22) at time k.

In the following, it is shown that the solution of (5.16) is also solution of the problem (5.13), and that the function

$$V(\Delta x(k)) = \Delta x^T(k)\mathbf{M}(k)\Delta x(k),\ \mathbf{M}(k) > 0, \tag{5.23}$$

where $\mathbf{M}(k) = \gamma(k)\boldsymbol{\Theta}(k)^{-1}$, with $\gamma(k)$ and $\boldsymbol{\Theta}(k)$ solutions of (5.16), is a Lyapunov function for (5.13) with the gain $F(k)$, of the control law $\Delta u(k) = F(k)\Delta x(k)$, given by

$$F(k) = \boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}. \tag{5.24}$$

- The inequality (5.17) is equivalent to $V(\Delta x(k)) \leq \gamma(k)$. Indeed, by expanding (5.17) by means of the Schur complement and letting $\boldsymbol{\Theta}(k) = \gamma(k)\mathbf{M}(k)^{-1}$ in (5.17), the following holds

$$\begin{aligned} 1 - \Delta x(k)^T\boldsymbol{\Theta}(k)^{-1}\Delta x(k) &\geq 0, \\ 1 - \Delta x(k)^T\gamma(k)^{-1}\boldsymbol{M}(k)\Delta x(k) &\geq 0, \\ \Delta x(k)^T\gamma(k)^{-1}\boldsymbol{M}(k)\Delta x(k) &\leq 1, \\ \Delta x(k)^T\boldsymbol{M}(k)\Delta x(k) &\leq \gamma(k), \\ V(\Delta x(k)) &\leq \gamma(k). \end{aligned} \tag{5.25}$$

  Therefore, the right side of (5.14) is fulfilled by (5.17).

- The inequalities (5.18) and (5.19) are equivalent to that

$$\max_{[\mathbf{A}(\lambda(k+i))\ \mathbf{B}]\in\mathcal{C},\, i\geq 0} J(k) \leq V(\Delta x(k)). \tag{5.26}$$

The constraints (5.18) and (5.19) also guarantee that the function (5.23), with $\mathbf{M}(k) = \gamma(k)\mathbf{\Theta}(k)^{-1}$, is a quadratic Lyapunov function for the problem (5.13) for any $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$, $i \geq 0$, with $\Delta u(k) = F(k)\Delta x(k)$. Indeed, it can be shown (see Appendix B.2) that the inequalities (5.18) and (5.19) are equivalent to

$$\begin{aligned} V(\Delta x(k+i+1)) - V(\Delta x(k+i)) \leq \\ - \Big(\Delta x(k+i)^T \mathbf{Q} \Delta x(k+i) + \Delta u(k+i)^T \mathbf{R} \Delta u(k+i)\Big), \end{aligned} \tag{5.27}$$

for all $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$, $i \geq 0$, with $V(\Delta x(k))$ given by (5.23) and the gain $F(k)$, of the control law $\Delta u(k) = F(k)\Delta x(k)$, given by (5.24). The expression (5.27), is used in the following to show, firstly, that $V(\Delta x(k))$ in (5.23) is a Lyapunov function, and secondly, that (5.26) is achieved.

1. Since $\mathbf{Q} > 0$ and $\mathbf{R} > 0$, the right side of (5.27) is always negative. Thus, $V(\Delta x(k))$, is strictly decreasing for $i \geq 0$. Besides, from the definition of $V(\Delta x(k))$, in (5.23), $V(\Delta x(k) = 0) = 0$. Therefore, inequalities (5.18) and (5.19) guarantee that $V(\Delta x(k))$, as defined in (5.23) with $\mathbf{M}(k) = \gamma(k)\mathbf{\Theta}(k)^{-1}$, is a robust quadratic Lyapunov function for (5.13).
2. It can be shown (see Appendix B.3) that (5.27) is equivalent to

$$J(k) \leq V(\Delta x(k)), \tag{5.28}$$

which is satisfied for all $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$, $i \geq 0$. In particular,

$$\max_{[\mathbf{A}(\lambda(k+i))\ \mathbf{B}]\in\mathcal{C},\, i\geq 0} J(k) \leq V(\Delta x(k)). \tag{5.29}$$

From items 1 and 2, since the scalar $\gamma(k)$ is an upper bound for $J(k)$, the

control law solution of (5.16), i.e.

$$\Delta u(k) = \mathbf{\Lambda}(k)\mathbf{\Theta}(k)^{-1}\Delta x(k), \tag{5.30}$$

stabilises the system (5.1) to the origin in a robust asymptotic way.

- The matrix inequality (5.20) guarantees that (5.15) satisfies $\Delta u(k) \in U_\Delta$. Indeed, it can be shown (see Appendix B.4) that (5.20) is equivalent to

$$(\Delta u(k))^T \Delta u(k) \leq \Delta u_{max}^2. \tag{5.31}$$

- The constraints (5.21) and (5.22), guarantee that the symmetric constraints for the output are fulfilled, i.e. $\Delta y(k+i) \in Y_\Delta,\ \forall i \geq 1$. It is possible to shown (see Appendix B.5) that (5.21) and (5.22) are equivalent to

$$(\Delta y(k+i))^T \Delta y(k+i) \leq \Delta y_{max}^2,\ \forall i \geq 1. \tag{5.32}$$

It is worth pointing out that, the recursive feasibility of the problem (5.16) is proved in [42]. Therefore, the solutions computed at time k for the state $\Delta x(k)$ will be solutions for $\Delta x(k+i),\ i \geq 0$, as well.

The problem (5.16) involves at least $2 + 2p_{\mathcal{C}}$ LMIs constraints, with $p_{\mathcal{C}}$ the number of vertices in the polytope $\mathcal{C}$. And yet, the optimization problem (5.16) must be solved on-line at each time k. Hence, the problem (5.16) is not suitable for real-time applications. To handle this problem, [68] proposes the following off-line approach, here particularized to achieve the tracking task.

5.3.1 Explicit MPC for LPV systems with off-line computation of LMIs

The key idea of the explicit approach is to grid the state-space and to solve the problem (5.16) off-line for a finite number of the elements of the gridded state-space. The gains that are solutions of (5.16) are saved in look-up tables and are used on-line to steer $\Delta x(k)$ to zero. As it will be shown later, asymptotically stable invariant sets will guarantee that $\Delta x(k)$ is asymptotically stabilized to

zero, even for the points $\Delta x(k)$ which do not correspond to samples of the gridded state-space.

In Section 5.3.1.1, the concept of asymptotically stable invariant ellipsoids for (5.16) is introduced. Next, in Section 5.3.1.2, the principle and the algorithm of the explicit MPC are described.

5.3.1.1 Asymptotically stable invariant ellipsoids

Define the set $\xi(k)$ as the set of the states that verify (5.17), i.e.

$$\xi(k) = \{\Delta x(k) \in \mathbb{R}^2 : \Delta x(k)^T \boldsymbol{\Theta}(k)^{-1} \Delta x(k) \leq 1\}, \tag{5.33}$$

with $\boldsymbol{\Theta}(k)^{-1}$ being the solution of the optimization problem (5.16) at time k. Consider, at time $k+i$, a non-zero state $\Delta x(k+i)$ that is already in $\xi(k+i)$, i.e.

$$\Delta x(k+i)^T \boldsymbol{\Theta}(k+i)^{-1} \Delta x(k+i) \leq 1. \tag{5.34}$$

If the feedback control law $\Delta u(k+i) = \boldsymbol{\Lambda}(k+i)\boldsymbol{\Theta}(k+i)^{-1}\Delta x(k+i)$ (with $\boldsymbol{\Lambda}(k+i)$ and $\boldsymbol{\Theta}(k+i)$ the solutions of (5.16) at $k+i$) is applied to $\Delta x(k+i)$, then in virtue of the dynamics (5.1), it holds that

$$\Delta x(k+i+1) = \left(\mathbf{A}(\lambda(k+i)) + \mathbf{B}\boldsymbol{\Lambda}(k+i)\boldsymbol{\Theta}(k+i)^{-1}\right)\Delta x(k+i). \tag{5.35}$$

In [42, 68], it is demonstrated that from the recursive feasibility features of problem (5.16), the state (5.35) satisfies all the LMIs constraints (5.17)-(5.22) computed at time $k+i$, in particular

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}(k+i)^{-1} \Delta x(k+i+1) \leq 1. \tag{5.36}$$

Therefore, $\Delta x(k+i+1)$, given by (5.35), accomplishes that $\Delta x(k+i+1) \in \xi(k+i)$, $i \geq 0$, if $\Delta x(k+i) \in \xi(k+i)$. Additionally, since the control law $\Delta u(k) = \boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\Delta x(k)$ asymptotically stabilises the system (2.40) to the

origin, then

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}(k+i)^{-1} \Delta x(k+i+1) < \Delta x(k+i)^T \boldsymbol{\Theta}(k+i)^{-1} \Delta x(k+i), \ i \geq 0, \tag{5.37}$$

and thus (5.36) satisfies

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}(k+i)^{-1} \Delta x(k+i+1) < \Delta x(k+i)^T \boldsymbol{\Theta}(k+i)^{-1} \Delta x(k+i) \leq 1, \ i \geq 0. \tag{5.38}$$

From Definition 1, in Section 3.3, it is clear that for every k, the set $\xi(k)$ is an asymptotically stable invariant set [42, 68]. Since (5.33) describes an ellipsoid, then $\xi(k)$ is an ellipsoidal invariant set for the system (5.1) under the closed-loop state feedback control $\Delta u(k) = \boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\Delta x(k)$. Having being introduced the concept of asymptotically stable invariant ellipsoid for the problem (5.16), the algorithm of the explicit MPC with off-line computation of LMIs can be now presented.

5.3.1.2 The explicit MPC algorithm

Solving (5.16) off-line for a finite gridding of the state-space, i.e. for the set $\Delta\check{x} = \{\Delta x^{(1)}, \Delta x^{(2)}, \cdots, \Delta x^{(j)}\}$, gives the corresponding sets $\check{\boldsymbol{\Theta}} = \{\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \cdots, \boldsymbol{\Theta}^{(j)}\}$ and $\check{\boldsymbol{\Lambda}} = \{\boldsymbol{\Lambda}^{(1)}, \boldsymbol{\Lambda}^{(2)}, \cdots, \boldsymbol{\Lambda}^{(j)}\}$, where $\boldsymbol{\Theta}^{(i)}$ and $\boldsymbol{\Lambda}^{(i)}$ are the solutions of (5.16) for $\Delta x(k) = \Delta x^{(i)}$. The sets $\check{\boldsymbol{\Theta}}$ and $\check{\boldsymbol{\Lambda}}$, are saved in look-up tables. In virtue of (5.33), for each $\boldsymbol{\Theta}^{(i)}$ a corresponding invariant ellipsoidal set $\xi^{(i)}$ can be obtained. Thus, a set of concentric ellipsoidal invariant sets, i.e. $\check{\xi} = \{\xi^{(1)}, \xi^{(2)}, \cdots, \xi^{(j)}\}$, (see Fig. 5.1) is implicitly given by $\check{\boldsymbol{\Theta}}$.

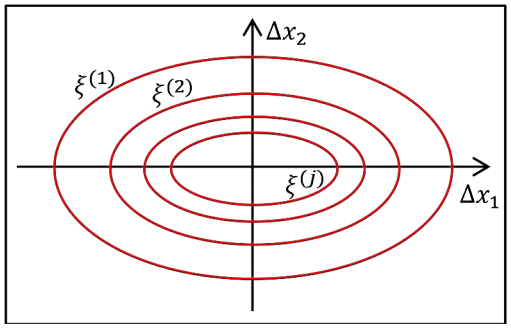


Figure 5.1: Example of set of concentric ellipsoids for a state-space of dimension 2.

On-line, at each time k, the smallest ellipsoid $\xi^{(i)} \in \check{\xi}$ that contains the actual state $\Delta x(k)$ (see Fig. 5.2) is chosen. Next, the closed-loop control $\Delta u(k) = \Lambda^{(i)} \left(\mathbf{\Theta}^{(i)}\right)^{-1} \Delta x(k)$ is applied, where the gains $\mathbf{\Lambda}^{(i)}$ and $\left(\mathbf{\Theta}^{(i)}\right)^{-1}$ of the look-up table correspond to the ellipsoid $\xi^{(i)}$.

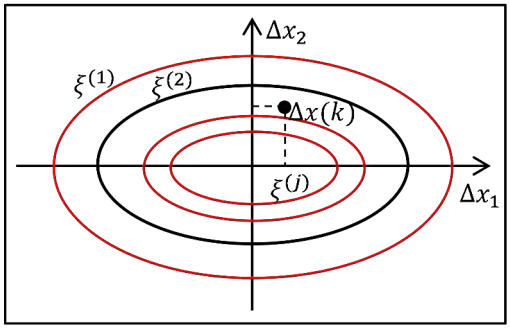


Figure 5.2: The smallest ellipsoid that contains the actual state $\Delta x(k)$ is $\xi^{(2)}$ and is depicted in black.

Remark 5 *Notice that for all the states $\Delta x(k) \in \xi(k)$, given by (5.33), all the LMIs constraints (5.17)-(5.22) are feasible, but not necessarily optimal [68]. However, although $\Delta u(k) = \Lambda^{(i)} \left(\mathbf{\Theta}^{(i)}\right)^{-1} \Delta x(k)$ is not necessarily optimal, it still being an admissible solution for (5.16), since (5.17)-(5.22) are satisifed for $\Delta x(k)$ if $\Delta x(k) \in \xi^{(i)}$.*

The algorithm that sums up the overall explicit approach is detailed in the following.

Algorithm 3

- Off-line steps

  1. For practical purposes, it is convenient to consider the gridding of only one component Δx_i of Δx, letting the remaining components constant. Choose a set of values $\Delta \check{x}_i = \{\Delta x_i^{(1)}, \Delta x_i^{(2)}, \cdots, \Delta x_i^{(j)}\}$, as far from the origin as possible while preserving the feasibility of the problem.

2. Solve (5.16) for each element $\Delta x^{(j)}$ in the set $\Delta\check{x}$ of states, and save in a look-up table the corresponding sets $\check{\mathbf{\Theta}} = \{\mathbf{\Theta}^{(1)}, \mathbf{\Theta}^{(2)}, \cdots, \mathbf{\Theta}^{(j)}\}$ and $\check{\mathbf{\Lambda}} = \{\mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \cdots, \mathbf{\Lambda}^{(j)}\}$.

The closed-loop control consists in the following real-time strategy performed on-line.

- On-line steps

1. Measure $\Delta x(k) = [\Delta x_1(k),\, \Delta x_2(k)]^T$ at time k.
2. Search in the look-up table the gain $\mathbf{\Theta}^{(i)}$ such that $\xi^{(i)} = \{\Delta x(k) \in \mathbb{R}^2 : \Delta x(k)^T \left(\mathbf{\Theta}^{(i)}\right)^{-1} \Delta x(k) \leq 1\}$ is the smallest invariant ellipsoid that contains $\Delta x(k)$.
3. Apply $\Delta u(k) = \mathbf{\Lambda}^{(i)} \left(\mathbf{\Theta}^{(i)}\right)^{-1} \Delta x(k)$.
4. Return to Step 1.

In the next section, Algorithm 3 is applied to the *Vir'volt* vehicle described in Section 2.2.

5.4 Robust MPC for LPV systems: application to the Vir'volt vehicle

Consider the LPV representation (5.1) of the dynamics of the tracking error $\Delta x(k)$, and the corresponding tracking problem (5.16) with weighting matrices $\mathbf{Q} = [0\ 0; 0\ 1]^T$ and $\mathbf{R} = 1$. The tracking constraints are imposed as follows

$$-2\text{km/h} \leq \Delta y(k) \leq 2\text{km/h and } 0.5\text{A} \leq \Delta u(k) \leq 0.5\text{A}. \tag{5.39}$$

The driving strategy to be tracked is depicted in Fig. 2.8. The tracking task is performed by applying the closed loop control law to the nonlinear discrete-dynamics (2.22). A mass variation of 20% is considered within (2.22) in order to asses the robustness to small disturbances. The sampling period is $T_s = 0.2$s.

5.4.1 Robust MPC for LPV systems with on-line computation of LMIs

The tracking task is performed by fully solving on-line the problem (5.16) at each execution time, and applying $\Delta u(k) = \mathbf{\Lambda}(k)\mathbf{\Theta}(k)^{-1}\Delta x(k)$ (see Fig. 5.3).

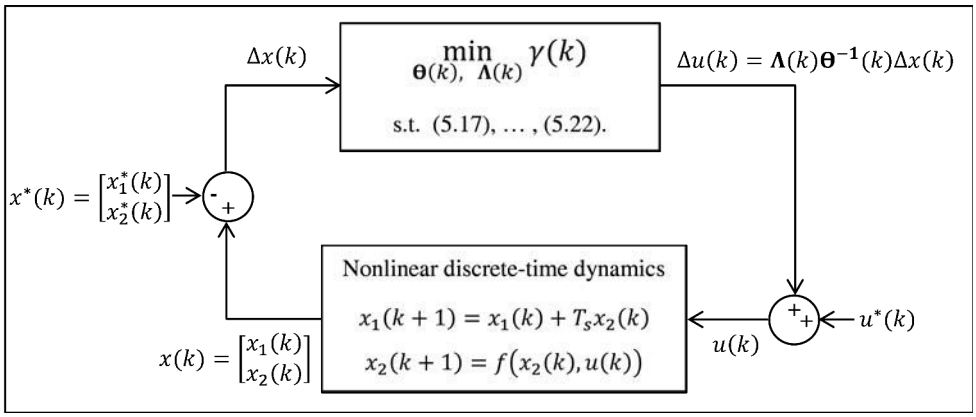


Figure 5.3: Closed-loop implementation of the robust MPC for LPV systems approach of [42]. The function that describes the velocity $x_2(k+1) = f(x_2(k), u(k))$ is given by (2.22).

The tracking of the velocity $x_2^*(k)$ according to the actual position of the vehicle is depicted in Fig. 5.4 and 5.5, in black solid line.

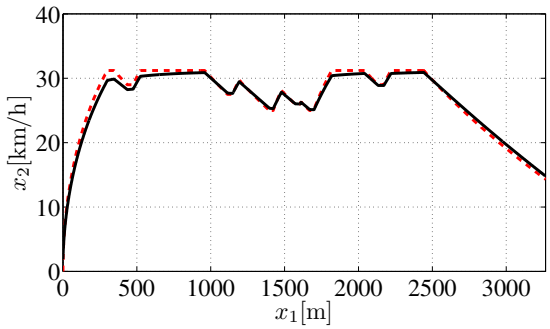


Figure 5.4: Closed-loop tracking response with a mass variation of 20%. Red dashed line: reference $x_2^*(k)$. Black solid line: tracking response of $x_2(k)$.

The difference between the actual nonlinear velocity $x_2(k)$ and the target state $x^*(k)$ at time k, i.e. the tracking error $\Delta x_2(k) = x_2(k) - x_2^*(k)$, is depicted in Fig. 5.6, in solid black line, with respect to the vehicle position.

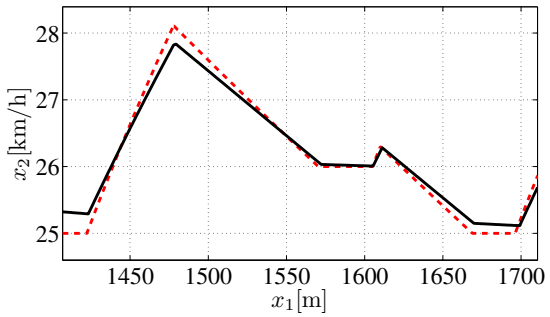


Figure 5.5: Detail of the closed-loop tracking response of Fig. 5.4.

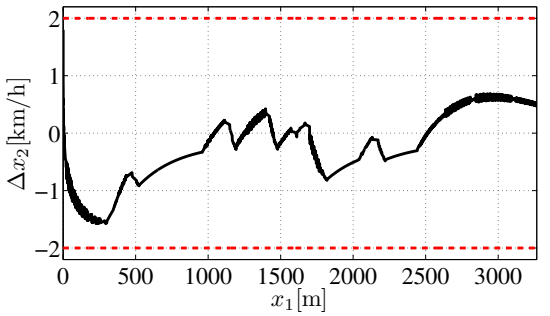


Figure 5.6: Closed-loop tracking error $\Delta x_2(k)$ with a mass variation of 20%. Red dashed lines: $\Delta x_{2min}(k)$ and $\Delta x_{2max}(k)$. Solid black line: tracking error $\Delta x_2(k)$.

In Fig. 5.7, in solid black line, the input for the nonlinear tracking $\Delta u(k) = u(k) - u^*(k)$ is depicted with respect to the vehicle position.

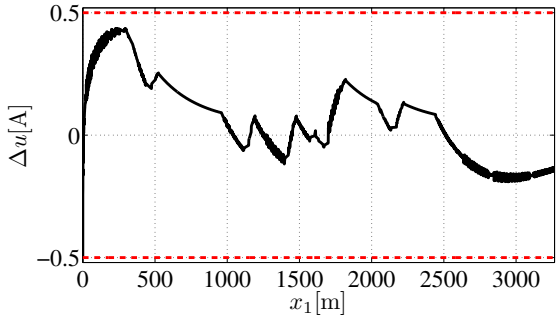


Figure 5.7: Closed-loop tracking error $\Delta u(k)$ with a mass variation of 20%. Red dashed lines: $\Delta u_{min}(k)$ and $\Delta u_{max}(k)$. Solid black line: tracking input $\Delta u(k)$.

It can be observed in Fig. 5.4-5.7 that the tracking task is well achieved, despite the mass variation. The tracking task is achieved while the constraints are satisfied. However, the aforementioned MPC strategy that consists in fully solving on-line the problem (5.16) at each execution time, is not compatible with the intended real-time application, because the problem (5.16) involves the on-line solution of six LMIs. Thus, as proposed in Section 5.3.1, the problem (5.16) is tackled with Algorithm 3 which does not require to solve LMIs on-line.

5.4.2 Explicit MPC for LPV systems with off-line computation of LMIs

Consider Algorithm 3 described in Section 5.3.1, with matrices $\mathbf{Q}$ and $\mathbf{R}$ as stated above, and constraints Δy_{max} and Δu_{max} as defined in (5.39). Algorithm 3 is applied to *Vir'volt* dynamics as depicted in the block digram of Fig. 5.8.

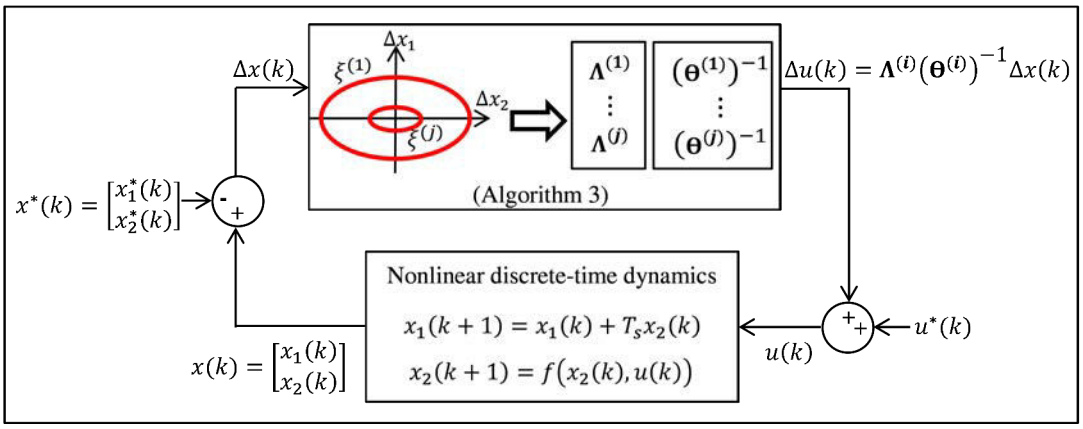


Figure 5.8: Closed-loop implementation of the explicit approach of the robust MPC for LPV systems (Algorithm 3) [68] with $j = 9$. The function that describes the velocity $x_2(k+1) = f(x_2(k), u(k))$ is given by (2.22).

The procedure performed to apply Algorithm 3 to the *Vir'volt* dynamics is described in the following.

- Off-line steps

  1. The gridding over the state Δx is chosen by making $\Delta x_1 = 0$ and

$$\Delta x_2 = \{1.80\text{km/h},\ 1.43\text{km/h},\ 0.90\text{km/h},\ 0.57\text{km/h},\ 0.36\text{km/h},\ 0.22\text{km/h},\ 0.14\text{km/h},\ 0.09\text{km/h},\ 0.03\text{km/h}\}, \tag{5.40}$$

     which is equivalent to

$$\Delta x_2 = \{0.5\text{m/s},\ 0.39\text{m/s},\ 0.25\text{m/s},\ 0.15\text{m/s},\ 0.1\text{m/s},\ 0.06\text{m/s},\ 0.03\text{m/s},\ 0.02\text{m/s},\ 0.01\text{m/s}\}. \tag{5.41}$$

  2. For each element in (5.41), i.e. $\Delta x_2^{(i)},\ i = 1, ..., 9$, and $\Delta x_1^{(i)} = 0,\ i = 1, ..., 9$, the problem (5.16) is solved off-line. The resulting $\mathbf{\Lambda}^{(i)}$ and $\mathbf{\Theta}^{(i)}$, $i = 1, ..., 9$, are saved in look-up tables. Each value $\mathbf{\Theta}^{(i)}$, $i = 1, ..., 9$, defines an invariant ellipsoid $\xi^{(i)}$ given by (5.33). The different ellipsoids $\xi^{(i)},\ i = 1, ..., 9$, are depicted in Fig. 5.9.

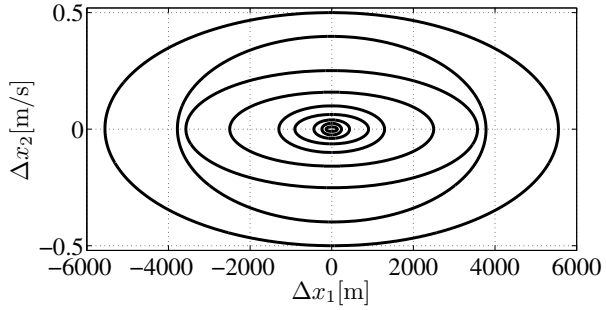


Figure 5.9: Invariant ellipsoids.

- On-line steps

  1.-2. At each time k, the smallest ellipsoid that contains the actual state $\Delta x(k)$, is used to select the corresponding $\mathbf{\Lambda}^{(i)}$ and $\left(\mathbf{\Theta}^{(i)}\right)^{-1}$.

3.-4. The closed loop control $\Delta u(k) = \mathbf{\Lambda}^{(i)} \left(\mathbf{\Theta}^{(i)}\right)^{-1} \Delta x(k)$ is applied to the nonlinear discrete-dynamics (2.22) that has a mass variation of 20% (see Fig. 5.8). The execution time is $T_s = 0.2$s.

The tracking response of Algorithm 3 is depicted, in green dashed line, in Fig. 5.10 and Fig. 5.11, for the tracking of the velocity $x_2^*(k)$ with respect to the actual position of the vehicle. The tracking error $\Delta x_2(k)$, is depicted in Fig. 5.12, in green dashed line, with respect to the vehicle position. In Fig. 5.13, the input for the nonlinear tracking $\Delta u(k)$ is also depicted in green dashed line.

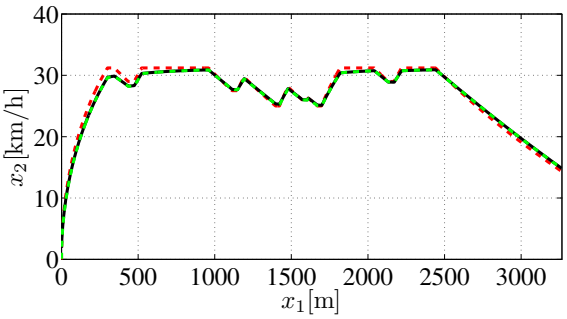


Figure 5.10: Closed-loop tracking response with a mass variation of 20%. Red dashed line: reference $x_2^*(k)$. Black solid line: tracking response of the on-line solution of the problem (5.16) at each execution time. Green dashed line: tracking response of the Algorithm 3.

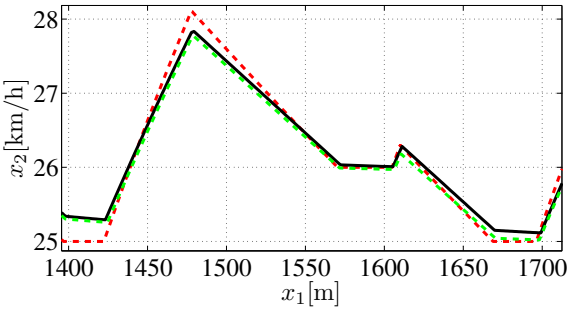


Figure 5.11: Detail of the closed-loop tracking response of Fig. 5.10.

From Fig. 5.10-5.13, it can be observed that the tracking task is achieved while the constraints are satisfied. The performances of Algorithm 3 are acceptable

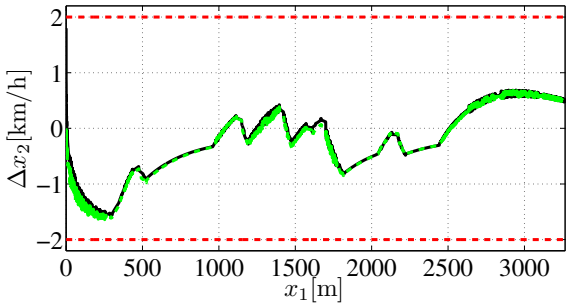


Figure 5.12: Closed-loop tracking error $\Delta x_2(k)$ with a mass variation of 20%. Red dashed lines: $\Delta x_{2min}(k)$ and $\Delta x_{2max}(k)$. Solid black line: tracking error $\Delta x_2(k)$ of the on-line solution of the problem (5.16) at each execution time. Green dashed line: tracking error $\Delta x_2(k)$ of the Algorithm 3.

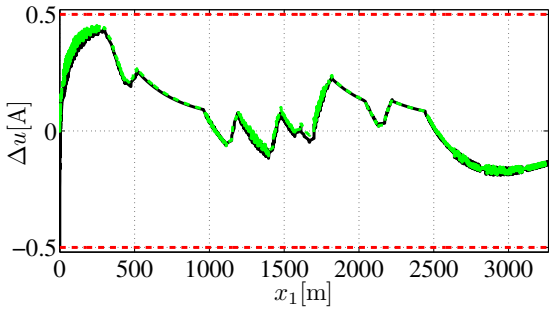


Figure 5.13: Closed-loop tracking error $\Delta u(k)$ with a mass variation of 20%. Red dashed lines: $\Delta u_{min}(k)$ and $\Delta u_{max}(k)$. Solid black line: tracking input $\Delta u(k)$ of the on-line solution of the problem (5.16) at each execution time. Green dashed line: tracking input $\Delta u(k)$ of the Algorithm 3.

even for a mass variation of 20%. It can be also observed that the performance of Algorithm 3, based on invariant ellipsoids computed off-line (see Fig. 5.8), is similar to that one of the solution of the problem (5.16) fully computed on-line (see Fig. 5.3).

5.4.3 Comparison of the fully on-line MPC and explicit MPC

The computational resources (computation time and memory) used on-line by the fully on-line MPC approach (fully on-line computation of problem (5.16) in

Section 5.3) and the explicit MPC (Algorithm 3 in Section 5.3.1), are related in Table 5.1.

Table 5.1: On-line computational resources of the fully on-line MPC and explicit MPC.

Method	**Computation time**	**Memory used**
Fully on-line computation of problem (5.16) in Section 5.3	21.4ms	4768Bytes
Algorithm 3: explicit MPC based on invariant ellipsoids	0.52ms	824Bytes

It can be observed that the computational time and the memory resources of Algorithm 3, are much more smaller than those required to fully solve on-line the problem (5.16) at each execution time. This result was expected since no LMIs is solved on-line in Algorithm 3. Therefore, Algorithm 3 appears as a good candidate to be implemented in real-time. However, Algorithm 3 does not includes the parameter $\lambda(k)$ in the control law, thus it is conservative. It would be interesting to consider other MPC strategies, also suitable for real-time implementation, that include the parameters $\lambda(k)$ in a explicit way, so that conservativeness can be reduced. To this end, in the following section, an alternative MPC strategy for LPV systems is considered. This strategy is based on a Parameter Dependent Lyapunov Function (PDLF) to include the parameter $\lambda(k)$ explicitly in the control law.

5.5 MPC for LPV systems with Parameter dependent Lyapunov function

Despite of the availability of the time-varying parameter $\lambda(k)$ in (5.1) at every time k, it is neither included in the feedback control law (5.24), nor in the robust Lyapunov function (5.23) used to stabilize the system. This makes the approach highly conservative since, at each time k, only one Lyapunov function and a unique corresponding gain are used to stabilise all the possible realizations of the polytopic LPV system, no matters the actual value of the parameter $\lambda(k)$ [66, 22].

The same remark applies for the off-line approach in [68], described in Algorithm 3.

Regarding this issue, several approaches have been proposed to reduce the conservativeness of the MPC problem for LPV systems. In particular, the works of [66, 22], that explicitly include the parameter in the Lyapunov function by introducing a Parameter Dependent Lyapunov Function (PDLF) in (5.23). Consequently, the parameter $\lambda(k)$ appears explicitly in the feedback control law $\Delta u(k) = F(k)\Delta x(k)$ with $F(k) = F(\lambda(k))$. The PDLF reads

$$V(\Delta x(k), \lambda(k)) = \Delta x^T(k)\mathbf{M}(\lambda(k))\Delta x(k),\ \mathbf{M}(\lambda(k)) > 0, \tag{5.42}$$

with

$$\mathbf{M}(\lambda(k)) = f_1(\lambda(k))\mathbf{M}_1(k) + f_2(\lambda(k))\mathbf{M}_2(k), \tag{5.43}$$

being f_1 and f_2 defined as in (2.45), and

$$\mathbf{M}_j(k) = \gamma(k)\mathbf{\Theta}_j(k)^{-1},\ j = 1, 2. \tag{5.44}$$

where $\gamma(k)$ and $\mathbf{\Theta}_j(k)$ are solutions of the LMIs convex optimization problem presented in the following.

Consider the tracking costs $\mathbf{Q}$ and $\mathbf{R}$, and the closed loop problem (5.13) for the polytopic LPV system (5.1), subject to tracking constraints X_Δ and U_Δ given by (5.7). According to [66], this problem can be properly solved using the parameter dependent feedback control law $F(k) = F(\lambda(k))$ and the PDLF (5.42), as follows

$$\min_{\mathbf{\Theta}_{1,2}(k),\ \mathbf{G}_{1,2}(k),\ \mathbf{\Lambda}_{1,2}(k)} \gamma(k), \tag{5.45}$$

s.t.

$$\begin{bmatrix} 1 & \star \\ \Delta x(k) & \mathbf{\Theta}_1(k) \end{bmatrix} \geq 0,\ \mathbf{\Theta}_1(k) > 0, \tag{5.46}$$

$$\begin{bmatrix} 1 & \star \\ \Delta x(k) & \mathbf{\Theta}_2(k) \end{bmatrix} \geq 0,\ \mathbf{\Theta}_2(k) > 0, \tag{5.47}$$

$$\begin{bmatrix} \mathbf{G}_1(k) + \mathbf{G}_1(k)^T - \mathbf{\Theta}_1(k) & \star & \star & \star \\ \mathbf{A}_1\mathbf{G}_1(k) + \mathbf{B}\mathbf{\Lambda}_1(k) & \mathbf{\Theta}_1(k) & \star & \star \\ \mathbf{Q}^{1/2}\mathbf{G}_1(k) & 0 & \gamma(k)I & \star \\ \mathbf{R}^{1/2}\mathbf{\Lambda}_1(k) & 0 & 0 & \gamma(k)I \end{bmatrix} \geq 0, \tag{5.48}$$

$$\begin{bmatrix} \mathbf{G}_1(k) + \mathbf{G}_1(k)^T - \mathbf{\Theta}_1(k) & \star & \star & \star \\ \mathbf{A}_1\mathbf{G}_1(k) + \mathbf{B}\mathbf{\Lambda}_1(k) & \mathbf{\Theta}_2(k) & \star & \star \\ \mathbf{Q}^{1/2}\mathbf{G}_1(k) & 0 & \gamma(k)I & \star \\ \mathbf{R}^{1/2}\mathbf{\Lambda}_1(k) & 0 & 0 & \gamma(k)I \end{bmatrix} \geq 0, \tag{5.49}$$

$$\begin{bmatrix} \mathbf{G}_2(k) + \mathbf{G}_2(k)^T - \mathbf{\Theta}_2(k) & \star & \star & \star \\ \mathbf{A}_2\mathbf{G}_2(k) + \mathbf{B}\mathbf{\Lambda}_2(k) & \mathbf{\Theta}_1(k) & \star & \star \\ \mathbf{Q}^{1/2}\mathbf{G}_2(k) & 0 & \gamma(k)I & \star \\ \mathbf{R}^{1/2}\mathbf{\Lambda}_2(k) & 0 & 0 & \gamma(k)I \end{bmatrix} \geq 0, \tag{5.50}$$

$$\begin{bmatrix} \mathbf{G}_2(k) + \mathbf{G}_2(k)^T - \mathbf{\Theta}_2(k) & \star & \star & \star \\ \mathbf{A}_2\mathbf{G}_2(k) + \mathbf{B}\mathbf{\Lambda}_2(k) & \mathbf{\Theta}_2(k) & \star & \star \\ \mathbf{Q}^{1/2}\mathbf{G}_2(k) & 0 & \gamma(k)I & \star \\ \mathbf{R}^{1/2}\mathbf{\Lambda}_2(k) & 0 & 0 & \gamma(k)I \end{bmatrix} \geq 0, \tag{5.51}$$

$$\begin{bmatrix} \Delta u_{max}^2\mathbf{I} & \star \\ \mathbf{\Lambda}_1(k)^T & \mathbf{G}_1(k) + \mathbf{G}_1(k)^T - \mathbf{\Theta}_1(k) \end{bmatrix} \geq 0, \tag{5.52}$$

$$\begin{bmatrix} \Delta u_{max}^2\mathbf{I} & \star \\ \mathbf{\Lambda}_2(k)^T & \mathbf{G}_2(k) + \mathbf{G}_2(k)^T - \mathbf{\Theta}_2(k) \end{bmatrix} \geq 0, \tag{5.53}$$

$$\begin{bmatrix} \Delta y_{max}^2 \mathbf{I} & \star \\ \left(\mathbf{A}_1\mathbf{G}_1(k) + \mathbf{B}\mathbf{\Lambda}_1(k)\right)^T \mathbf{C}^T & \mathbf{G}_1(k) + \mathbf{G}_1(k)^T - \mathbf{\Theta}_1(k) \end{bmatrix} \geq 0, \tag{5.54}$$

$$\begin{bmatrix} \Delta y_{max}^2 \mathbf{I} & \star \\ \left(\mathbf{A}_2\mathbf{G}_2(k) + \mathbf{B}\mathbf{\Lambda}_2(k)\right)^T \mathbf{C}^T & \mathbf{G}_2(k) + \mathbf{G}_2(k)^T - \mathbf{\Theta}_2(k) \end{bmatrix} \geq 0, \tag{5.55}$$

Similarly that in (5.16), the symbol $\star$ denotes the corresponding block to rend each matrix symmetric. Additionally, $\mathbf{\Theta}_j(k) \in \mathbb{R}^{n\times n}$, $j = 1, 2$, are symmetric matrices, $\mathbf{G}_j(k) \in \mathbb{R}^{n\times n}$ and $\mathbf{\Lambda}_j(k) \in \mathbb{R}^{1\times n}$, $j = 1, 2$. As in (5.16), the gain $\gamma(k)$ and the matrices $\mathbf{\Theta}_j(k)$, $\mathbf{G}_j(k)$ and $\mathbf{\Lambda}_j(k)$, $j = 1, 2$, in (5.45) are denoted as dependent on time k to emphasize that they are computed at each execution time k.

The feedback control law that minimizes (5.45), is given by

$$\Delta u(k) = F(\lambda(k))\Delta x(k), \tag{5.56}$$

with

$$F(\lambda(k)) = f_1(\lambda(k))F_1(k) + f_2(\lambda(k))F_2(k), \tag{5.57}$$

where $f_j(\lambda(k))$, $j = 1, 2$, are given by (5.6), and

$$F_j(k) = \mathbf{\Lambda}_j(k)\mathbf{G}_j^{-1}(k), \ j = 1, 2. \tag{5.58}$$

- As it is demonstrated in [66], and following the same lines as in Section 5.3, the LMIs constraints (5.46)-(5.51) guarantee that the Lyapunov function $V(\Delta x(k), \lambda(k))$ in (5.42), is a strictly decreasing for any $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$, $i \geq 0$. Thus, the asymptotic stability of the feedback control law (5.56) is given.

- The constraints (5.52)-(5.53) guarantee that $\Delta u(k+i) \in U_\Delta, i \geq 0$. The constraints (5.54)-(5.55) guarantee that $\Delta y(k+i) \in Y_\Delta, i \geq 1$ (see [66]).

The recursive feasibility of the problem (5.45) is proved in [66]. The recursive feasibility stands that if the problem (5.45) is feasible for $\Delta x(k)$, then the feasible solutions computed at time k are also feasible solutions for all the future states

$\Delta x(k+i),\ i \geq 0$, as long as the future parameter $\lambda(k+i)$ measured at time $k+i,\ i \geq 0$, makes $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C},\ i \geq 0$.

Notice that in the problem (5.45), there are at least $3p_{\mathcal{C}} + p_{\mathcal{C}}^2$ LMIs constraints to solve on-line, with $p_{\mathcal{C}}$ the number of vertices in the polytope $\mathcal{C}$. The number of LMIs constraints involved in (5.45) is higher than the number of LMIs constraints involved in (5.16) which is at least $2 + 2p_{\mathcal{C}}$ LMIs. Therefore, the computation time and the required memory to solve (5.45) are expected to be higher than those used to solve (5.16).

Having in mind the objective of a real-time implementation of the MPC strategy, and inspired by Algorithm 3 of [68] based in a quadratic Lyapunov function, in Section 5.6, a new explicit algorithm is developed for the problem (5.45). This explicit algorithm includes the time-varying parameter $\lambda(k)$, by resorting to the PDLF (5.42), in the control law (5.56), obtaining a parameter dependent control law. It also reduces the computational burden, since no on-line optimization of LMIs is performed on-line. In Sections 5.7.2 and 5.8, the proposed approach is applied to the *Vir'volt* vehicle.

5.6 Explicit MPC using the PDLF

The key idea of the explicit approach consists in solving off-line the problem (5.45) for a finite number of the elements of the gridded state-space. The gains $\mathbf{\Theta}_j(k)$, $\mathbf{G}_j(k)$ and $\mathbf{\Lambda}_j(k)$, $j = 1, 2$, that are solutions of (5.45) are saved in look-up tables. The look-up tables are used on-line to steer $\Delta x(k)$ to zero. Similarly that in Algorithm 3, asymptotically stable invariant sets, but not necessarily ellipsoids, will stabilize $\Delta x(k)$ to zero in an asymptotic way, even for the points $\Delta x(k)$ which do not correspond to samples of the gridded state-space.

5.6.1 The asymptotically stable invariant set

In this section, a similar procedure to the one developed in Section 5.3.1 is carried out. The objective is to demonstrate that the problem (5.45) admits a suitable invariant set under the control law (5.56). This invariant set, that is not necessarily an ellipsoid, conversely to the one found in Section 5.3.1.1, will be used to

develop an explicit algorithm of the problem (5.45).

Notice that the LMIs (5.46) and (5.47) define the sets $\xi_1(k)$ and $\xi_2(k)$ as follows

$$\xi_1(k) = \{\Delta x \in \mathbb{R}^2 : \Delta x(k)^T \boldsymbol{\Theta}_1(k)^{-1} \Delta x(k) \leq 1\}, \tag{5.59}$$

and

$$\xi_2(k) = \{\Delta x \in \mathbb{R}^2 : \Delta x(k)^T \boldsymbol{\Theta}_2(k)^{-1} \Delta x(k) \leq 1\}, \tag{5.60}$$

where $\boldsymbol{\Theta}_1(k)^{-1}$ and $\boldsymbol{\Theta}_2(k)^{-1}$ are solutions of the optimization problem (5.45) at time k. The states $\Delta x(k)$ that simultaneously belong to $\xi_1(k)$ and $\xi_2(k)$, i.e. the states $\Delta x(k)$ that belongs to the intersection between $\xi_1(k)$ and $\xi_2(k)$, satisfy all the LMIs constraints (5.46)-(5.55). The intersection between $\xi_1(k)$ and $\xi_2(k)$, denoted $\Omega_1(k)$, reads

$$\begin{aligned}
\Omega_1(k) &= \{\Delta x(k) \in \mathbb{R}^2 : \Delta x(k) \in \xi_1(k) \wedge \Delta x(k) \in \xi_2(k)\}, \\
&= \left\{\Delta x(k) \in \mathbb{R}^2 : \left(\Delta x(k)^T \boldsymbol{\Theta}_1(k)^{-1} \Delta x(k) \leq 1\right) \wedge \left(\Delta x(k)^T \boldsymbol{\Theta}_2(k)^{-1} \Delta x(k) \leq 1\right)\right\}, \\
&= \xi_1(k) \bigcap \xi_2(k).
\end{aligned} \tag{5.61}$$

Notice that although $\xi_1(k)$ and $\xi_2(k)$ describe ellipsoids, the intersection of ellipsoids is not necessarily an ellipsoid.

If at time $k+i$, the control (5.56) is applied to a non-zero state $\Delta x(k+i)$ that satisfies $\Delta x(k+i) \in \Omega_1(k+i)$, then in virtue of (5.1) the resulting state $\Delta x(k+i+1)$ reads

$$\begin{aligned}
\Delta x(k+i+1) &= \Big(\mathbf{A}(\lambda(k+i)) + \mathbf{B}F(\lambda(k+i))\Big)\Delta x(k+i), \\
&= \Big(\mathbf{A}(\lambda(k+i)) + \mathbf{B}\left(f_1(\lambda(k+i))F_1(k+i)\right. \\
&\qquad \left. + f_2(\lambda(k+i))F_2(k+i)\right)\Big)\Delta x(k+i), \\
&= \Big(\mathbf{A}(\lambda(k+i)) + \mathbf{B}\left(f_1(\lambda(k+i))\boldsymbol{\Lambda}_1(k+i)\mathbf{G}_1^{-1}(k+i)\right. \\
&\qquad \left. + f_2(\lambda(k+i))\boldsymbol{\Lambda}_2(k+i)\mathbf{G}_2^{-1}(k+i)\right)\Big)\Delta x(k+i),
\end{aligned} \tag{5.62}$$

with $\boldsymbol{\Lambda}_1(k+i), \mathbf{G}_1(k+i), \boldsymbol{\Lambda}_2(k+i)$ and $\mathbf{G}_2(k+i)$ solutions of (5.45) at time $k+i$.

Due to the recursively feasibility of the problem (5.45), if the state $\Delta x(k+i)$ satisfies all the LMIs constraints (5.46)-(5.55), computed at time $k+i$, then the state $\Delta x(k+i+1)$ also satisfies the constraints (5.46)-(5.55), computed at time $k+i$. In particular $\Delta x(k+i+1)$ satisfies (5.46), i.e.

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}_1(k+i)^{-1} \Delta x(k+i+1) \leq 1, \tag{5.63}$$

and

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}_2(k+i)^{-1} \Delta x(k+i+1) \leq 1, \tag{5.64}$$

with $\boldsymbol{\Theta}_1(k+i)$ and $\boldsymbol{\Theta}_2(k+i)$ computed at time $k+i$. Besides, the state feedback control law (5.56) asymptotic stabilizes the system to the origin, thus

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}_1(k+i)^{-1} \Delta x(k+i+1) < \Delta x(k+i)^T \boldsymbol{\Theta}_1(k+i)^{-1} \Delta x(k+i), \tag{5.65}$$

and

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}_2(k+i)^{-1} \Delta x(k+i+1) < \Delta x(k+i)^T \boldsymbol{\Theta}_2(k+i)^{-1} \Delta x(k+i). \tag{5.66}$$

Therefore,

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}_2(k+i)^{-1} \Delta x(k+i+1) < \Delta x(k+i)^T \boldsymbol{\Theta}_2(k+i)^{-1} \Delta x(k+i) \leq 1, \tag{5.67}$$

and

$$\Delta x(k+i+1)^T \boldsymbol{\Theta}_2(k+i)^{-1} \Delta x(k+i+1) < \Delta x(k+i)^T \boldsymbol{\Theta}_2(k+i)^{-1} \Delta x(k+i) \leq 1. \tag{5.68}$$

From (5.67) and (5.59) it holds that $\Delta x(k+i+1) \in \xi_1(k+i)$, $i \geq 0$, and from (5.68) and (5.60) it holds that $\Delta x(k+i+1) \in \xi_2(k+i)$, $i \geq 0$. Thus, using (5.61), it is obtained that

$$\Delta x(k+i+1) \in \Omega_1(k+i), \text{ if } \Delta x(k+i) \in \Omega_1(k+i), \ i \geq 0. \tag{5.69}$$

Therefore, the set $\Omega_1(k+i)$ is an asymptotically stable invariant set under the closed-loop state feedback control $\Delta u(k) = F(\lambda(k))\Delta x(k)$.

5.6.2 Algorithm of the explicit MPC based on PDLF

Similarly to Algorithm 3, if the problem (5.45) is solved off-line for a finite gridding of the state-space $\Delta\check{x} = \{\Delta x^{(1)}, \Delta x^{(2)}, \cdots, \Delta x^{(j)}\}$, the following corresponding sets are obtained

$$\begin{aligned}
\check{\boldsymbol{\Theta}}_1 &= \{\boldsymbol{\Theta}_1^{(1)}, \boldsymbol{\Theta}_1^{(2)}, \cdots, \boldsymbol{\Theta}_1^{(j)}\},\\
\check{\boldsymbol{\Theta}}_2 &= \{\boldsymbol{\Theta}_2^{(1)}, \boldsymbol{\Theta}_2^{(2)}, \cdots, \boldsymbol{\Theta}_1^{(j)}\},\\
\check{\boldsymbol{G}}_1 &= \{\boldsymbol{G}_1^{(1)}, \boldsymbol{G}_1^{(2)}, \cdots, \boldsymbol{G}_1^{(j)}\},\\
\check{\boldsymbol{G}}_2 &= \{\boldsymbol{G}_2^{(1)}, \boldsymbol{G}_2^{(2)}, \cdots, \mathbf{G}_2^{(j)}\},\\
\check{\boldsymbol{\Lambda}}_1 &= \{\boldsymbol{\Lambda}_1^{(1)}, \boldsymbol{\Lambda}_1^{(2)}, \cdots, \boldsymbol{\Lambda}_1^{(j)}\},\\
\check{\boldsymbol{\Lambda}}_2 &= \{\boldsymbol{\Lambda}_2^{(1)}, \boldsymbol{\Lambda}_2^{(2)}, \cdots, \boldsymbol{\Lambda}_2^{(j)}\},
\end{aligned} \tag{5.70}$$

where $\boldsymbol{\Theta}_{1,2}^{(i)}$, $\boldsymbol{G}_{1,2}^{(i)}$ and $\boldsymbol{\Lambda}_{1,2}^{(i)}$, $i = 1, ..., j$, are the solutions of (5.45) for $\Delta x(k) = \Delta x^{(i)}$, $i = 1, ..., j$. The sets $\check{\boldsymbol{\Theta}}_1$, $\check{\boldsymbol{\Theta}}_2$, $\check{\boldsymbol{G}}_1$, $\check{\boldsymbol{G}}_2$ $\check{\boldsymbol{\Lambda}}_1$ and $\check{\boldsymbol{\Lambda}}_2$, are saved in look-up tables.

Notice that from (5.59) and (5.60), two corresponding ellipsoidal sets $\xi_1^{(i)}$ and $\xi_2^{(i)}$ are obtained for each $\boldsymbol{\Theta}_1^{(i)}$ and $\boldsymbol{\Theta}_2^{(i)}$, $i = 1, ..., j$, respectively. Thus, from (5.61), for each $\boldsymbol{\Theta}_1^{(i)}$ and $\boldsymbol{\Theta}_2^{(i)}$, $i = 1, ..., j$, a corresponding invariant set $\Omega_1^{(i)}$ can be obtained. Therefore, the set $\check{\Omega}_1 = \{\Omega_1^{(1)}, \Omega_1^{(2)}, \cdots, \Omega_1^{(j)}\}$ of invariant sets, is obtained from the sets $\check{\boldsymbol{\Theta}}_1$ and $\check{\boldsymbol{\Theta}}_2$.

Remark 6 *Notice that for all the states $\Delta x(k) \in \xi_1(k) \bigcap \xi_2(k) = \Omega_1(k)$, the LMIs constraints (5.46)-(5.55) are feasible, but not necessarily optimal (the optimal solution of (5.45) at time k belongs to $\Omega_1(k)$, but not necessarily is the entire set). However, the control*

$$\Delta u(k) = \left(f_1(\lambda(k)) \boldsymbol{\Lambda}_1^{(i)} \left(\boldsymbol{G}_1^{(i)} \right)^{-1} + f_2(\lambda(k)) \boldsymbol{\Lambda}_2^{(i)} \left(\boldsymbol{G}_2^{(i)} \right)^{-1} \right) \Delta x(k) \tag{5.71}$$

still being an admissible solution for (5.45), since (5.46)-(5.55) are satisifed if $\Delta x(k) \in \Omega_1^{(i)}$.

The algorithm that sums up the overall explicit approach here proposed is detailed in the following.

Algorithm 4

- Off-line steps

  1. For practical purposes, it is convenient to consider the gridding of only one component Δx_i of Δx, letting the remaining components constant. Choose a set of values $\Delta\check{x}_i = \{\Delta x_i^{(1)}, \Delta x_i^{(2)}, \cdots, \Delta x_i^{(j)}\}$, as far from the origin as feasible the problem.
  2. Solve (5.45) for each value $\Delta x^{(i)}$, $i = 1, ..., j$, in the set of states $\Delta\check{x}$.
  3. Save in a look-up table the corresponding values of $\mathbf{\Theta}_1^{(i)}$, $\mathbf{\Theta}_2^{(i)}$, $\mathbf{G}_1^{(i)}$, $\mathbf{G}_2^{(i)}$ $\mathbf{\Lambda}_1^{(i)}$, and $\mathbf{\Lambda}_2^{(i)}$, $i = 1, ..., j$, that solve (5.45) for each $\Delta x_i^{(i)} \in \Delta\check{x}$, $i = 1, ..., j$.

- On-line steps

  1. Measure $\Delta x(k)$ at time k.
  2. Measure $\lambda(k)$ and find $f_j(\lambda(k))$, $j = 1, 2$, as given by (5.6), i.e.

$$f_1(\lambda(k)) = \frac{\lambda_{max} - \lambda(k)}{\lambda_{max} - \lambda_{min}} \text{ and } f_2(\lambda(k)) = \frac{\lambda(k) - \lambda_{min}}{\lambda_{max} - \lambda_{min}}. \tag{5.72}$$

  3. Search in the look-up table the gains $\mathbf{\Theta}_1^{(i)} \in \check{\mathbf{\Theta}}_1$ and $\mathbf{\Theta}_2^{(i)} \in \check{\mathbf{\Theta}}_2$ such that

$$\Omega_1^{(i)} = \left\{ \Delta x \in \mathbb{R}^2 : \left(\Delta x(k)^T \left(\mathbf{\Theta}_1^{(i)} \right)^{-1} \Delta x(k) \leq 1 \right) \wedge \left(\Delta x(k)^T \left(\mathbf{\Theta}_2^{(i)} \right)^{-1} \Delta x(k) \leq 1 \right) \right\}, \tag{5.73}$$

  is the smallest invariant set that contains $\Delta x(k)$.

4. Apply

$$\begin{aligned}\Delta u(k) &= F^{(i)}(\lambda(k))\Delta x(k),\\ &= \left(f_1(\lambda(k))F_1^{(i)} + f_2(\lambda(k))F_2^{(i)}\right)\Delta x(k),\\ &= \left(f_1(\lambda(k))\mathbf{\Lambda}_1^{(i)}\left(\mathbf{G}_1^{(i)}\right)^{-1} + f_2(\lambda(k))\mathbf{\Lambda}_2^{(i)}\left(\mathbf{G}_2^{(i)}\right)\right)\Delta x(k).\end{aligned} \tag{5.74}$$

5. Return to Step 1.

The proposed Algorithm 4, for the MPC using PDLF, is illustrated in the Sections 5.7.2 and 5.8.

5.7 Robust MPC for LPV systems with PDLF: application to the Vir'volt vehicle

In this section, the approach of [66], presented in Section 5.5, and the proposed Algorithm 4 of Section 5.6, are applied to the tracking problem of the *Vir'volt* vehicle tracking the optimal driving strategy. Then, they will be compared in terms of real-time capabilities. The conditions of the tracking task such as the tracking constraints, the sampling time, the optimal reference, etc., are the same that in Section 5.4.

Consider again the LPV representation (5.1) of the dynamics of the tracking error $\Delta x(k)$. The tracking constraints imposed to the problem (5.45), are the same than those imposed to the problem (5.16), i.e.

$$-2\text{km/h} \leq \Delta y(k) \leq 2\text{km/h and } -0.5\text{A} \leq \Delta u(k) \leq 0.5\text{A}. \tag{5.75}$$

The weighting matrices are $\mathbf{Q} = [0\ 0; 0\ 1]^T$ and $\mathbf{R} = 1$. The vehicle is required to follow the driving strategy of Fig. 2.8. The control loop is closed with the nonlinear discrete-dynamics (2.22) that has a mass variation of 20%, as in Section 5.4. The execution time is $T_s = 0.2$s.

5.7.1 Robust MPC with PDLF with on-line computation of LMIs

The tracking task is performed by fully solving the problem (5.45) on-line at each execution time, and applying $\Delta u(k) = F(\lambda(k))\Delta x(k)$ (see Fig. 5.14).

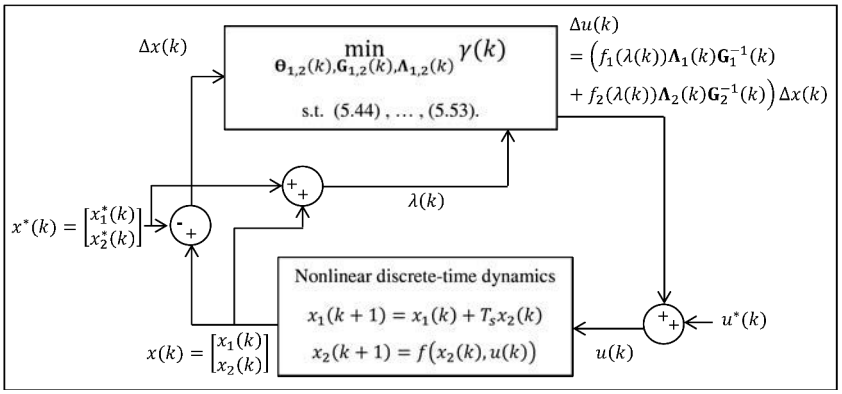


Figure 5.14: Closed-loop implementation of the MPC for LPV systems using PDLF [66]. The function that describes the velocity $x_2(k+1) = f(x_2(k), u(k))$ is given by (2.22).

The solution of the problem (5.45) is depicted in Fig. 5.15 and 5.16, in black solid line, for the tracking of the velocity $x_2^*(k)$ according to the actual position of the vehicle. The tracking error $\Delta x_2(k)$, is depicted in Fig. 5.17, in solid black line, with respect to the vehicle position. In Fig. 5.18, in solid black line, the input for the nonlinear tracking $\Delta u(k)$ is depicted. A mass variation of 20% has been imposed again to the vehicle dynamics. From Fig. 5.15-5.18 it can be observed that the tracking performance is acceptable despite the disturbance imposed in the mass. Besides, as will be shown later, the performances of the MPC strategy (5.45) are better than those of the strategy (5.16), since the first one is less conservative. Nevertheless, as in Section 5.4.1, the on-line solution of the problem (5.45) at each execution time, is not suitable for real-time applications. Therefore, in the following section, Algorithm 4 of Section 5.6, will be tested as an alternative to be implemented in real-time. Indeed, Algorithm 4 is less conservative that Algorithm 3 of Section 5.3.1, because the parameter $\lambda(k)$ is included explicitly in the control law, but requires less computational burden the control strategy (5.45), .

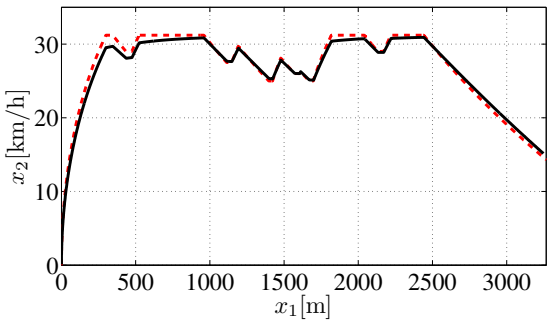


Figure 5.15: Closed-loop tracking response with a mass variation of 20%. Red dashed line: reference $x_2^*(k)$. Black solid line: tracking response of the on-line solution of the problem (5.45) at each execution time.

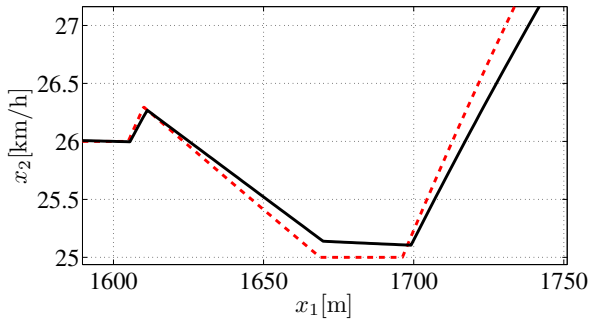


Figure 5.16: Detail of the closed-loop tracking response of Fig. 5.22.

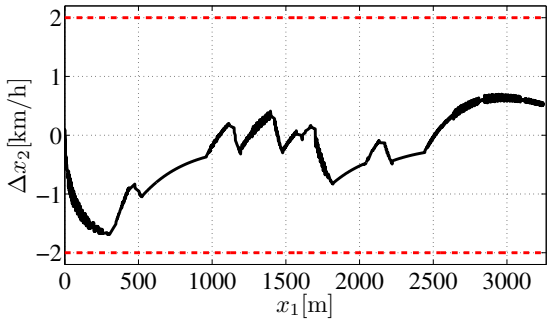


Figure 5.17: Closed-loop tracking error $\Delta x_2(k)$ with a mass variation of 20%. Red dashed lines: $\Delta x_{2min}(k)$ and $\Delta x_{2max}(k)$. Solid black line: tracking error $\Delta x_2(k)$ of the on-line solution of the problem (5.45) at each execution time.

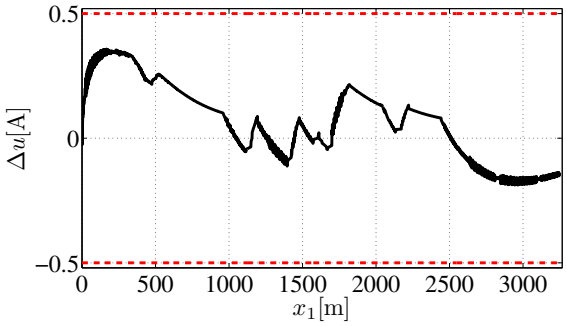


Figure 5.18: Closed-loop tracking error $\Delta u(k)$ with a mass variation of 20%. Red dashed lines: $\Delta u_{min}(k)$ and $\Delta u_{max}(k)$. Solid black line: tracking input $\Delta u(k)$ of the on-line solution of the problem (5.45) at each execution time.

5.7.2 Explicit MPC for LPV using PDLF

Consider now Algorithm 4 proposed in Section 5.6, with matrices $\mathbf{Q}$ and $\mathbf{R}$ as stated at the beginning of Section 5.7, and constraints Δy_{max} and Δu_{max} as defined in (5.75). It is applied to *Vir'volt* dynamics as depicted in the block diagram of Fig. 5.19.

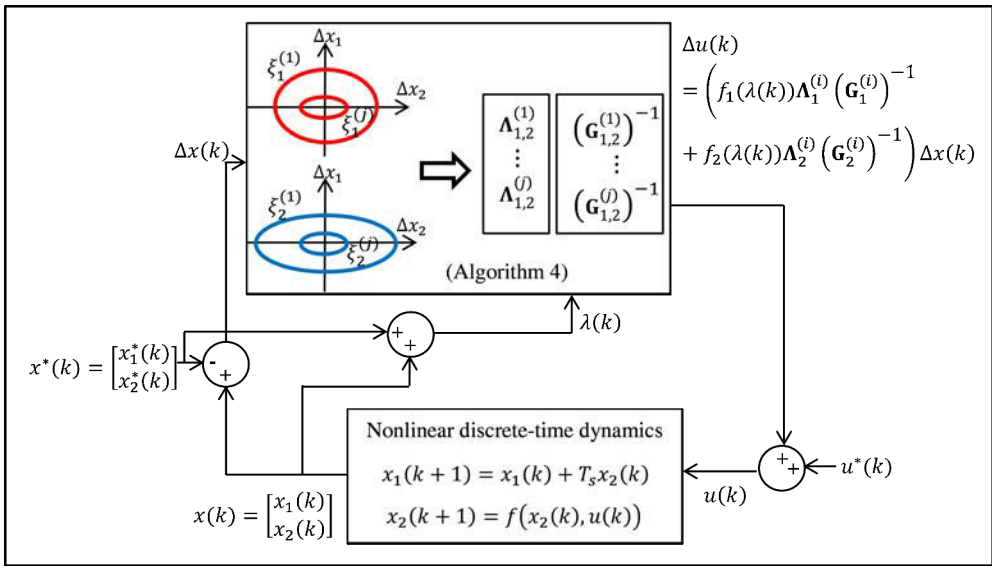


Figure 5.19: Closed-loop implementation of Algorithm 4. The function that describes the velocity $x_2(k+1) = f(x_2(k), u(k))$ is given by (2.22).

- Off-line steps

  1. The sampling over the state Δx is chosen by making $\Delta x_1 = 0$ and

$$\begin{aligned}\Delta x_2 = [&1.80\text{km/h},\ 1.43\text{km/h},\ 0.90\text{km/h},\ 0.57\text{km/h},\\ &0.14\text{km/h},\ 0.09\text{km/h},\ 0.05\text{km/h},\ 0.03\text{km/h}],\end{aligned} \tag{5.76}$$

     as in Section 5.4.

  2. For each element in (5.76), i.e. $\Delta x_2^{(i)}, i = 1, ..., 8$, the problem (5.45) is solved off-line, and the resulting $\mathbf{\Theta}_1^{(i)}$, $\mathbf{\Theta}_2^{(i)}$, $\mathbf{G}_1^{(i)}$, $\mathbf{G}_2^{(i)}$, $\mathbf{\Lambda}_1^{(i)}$, $\mathbf{\Lambda}_2^{(i)}$, $i = 1, ..., 8$, are saved in look-up tables.

  Each $\mathbf{\Theta}_1^{(i)}$ and $\mathbf{\Theta}_2^{(i)}$, $i = 1, ..., 8$, define the ellipsoids $\xi_1^{(i)}$ and $\xi_2^{(i)}$, given by (5.59) and (5.60), respectively. Those ellipsoids will be used on-line to find the corresponding invariant set $\Omega_1^{(i)}$ given by (5.61). In Fig. 5.20 and 5.21, are depicted the ellipsoids $\xi_1^{(i)}$ and $\xi_2^{(i)}$ such that $\Omega_1^{(i)} = \xi_1^{(i)} \bigcap \xi_2^{(i)}$, $i = 1, ..., 8$.

- On-line steps

  1.-3. At each time k, find the smallest invariant set $\Omega_1^{(i)}$ that contains the actual state $\Delta x(k)$. The index (i) is used to select the corresponding gains $\mathbf{\Lambda}_1^{(i)}$, $\mathbf{\Lambda}_2^{(i)}$, $\mathbf{G}_1^{(i)}$ and $\mathbf{G}_2^{(i)}$.

  4.-5. The control law

$$\Delta u(k) = \left(f_1(\lambda(k)) \mathbf{\Lambda}_1^{(i)} \left(\mathbf{G}_1^{(i)} \right)^{-1} + f_2(\lambda(k)) \mathbf{\Lambda}_2^{(i)} \left(\mathbf{G}_2^{(i)} \right)^{-1} \right) \Delta x(k), \tag{5.77}$$

  is applied to the system. The tracking task is performed by applying the closed-loop control to the nonlinear discrete-dynamics (2.22) that has a mass variation of 20%. The execution time is $T_s = 0.2$s.

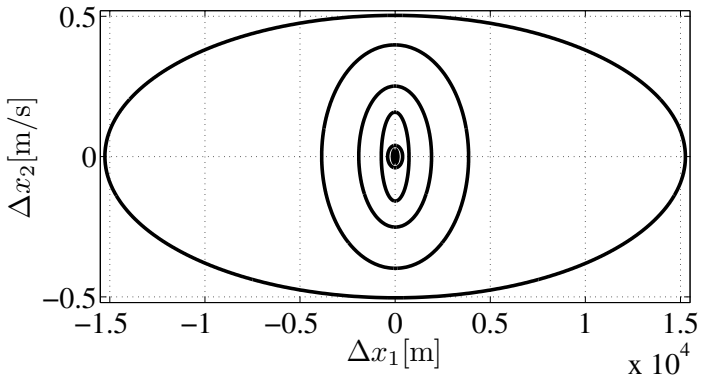


Figure 5.20: Ellipsoids $\xi_1^{(i)}$, $i = 1, ..., 8$.

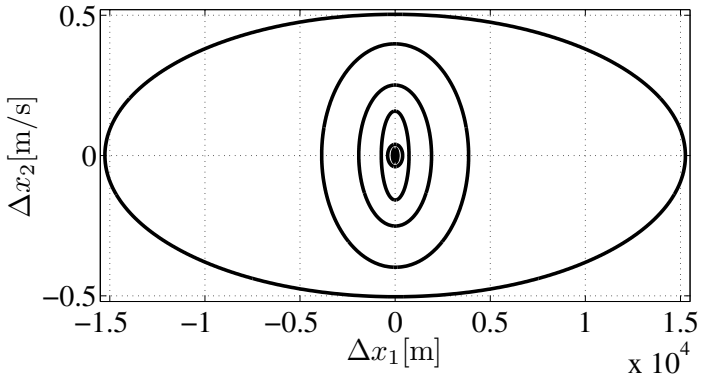


Figure 5.21: Ellipsoids $\xi_2^{(i)}$, $i = 1, ..., 8$.

The tracking response of the off-line approach is depicted, in cyan dashed line, in Fig. 5.22 and 5.23. The tracking error $\Delta x_2(k)$, is depicted in Fig. 5.24, and in Fig. 5.25, the input for the nonlinear tracking $\Delta u(k)$ is depicted in cyan dashed line. From Fig. 5.22-5.25, it can be observed that the performance of the proposed off-line approach (see Fig. 5.19), is similar to the performance of the on-line solution of (5.45)(see Fig. 5.14), despite the mass variation of 20%.

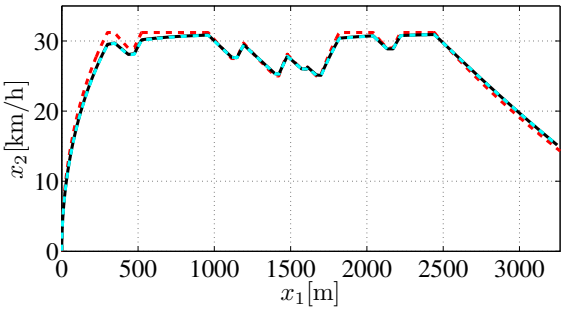


Figure 5.22: Closed-loop tracking response with a mass variation of 20%. Red dashed line: reference x_2^*. Black solid line: tracking response of the on-line solution of the problem (5.45) at each execution time. Cyan dashed line: tracking response of Algorithm 4.

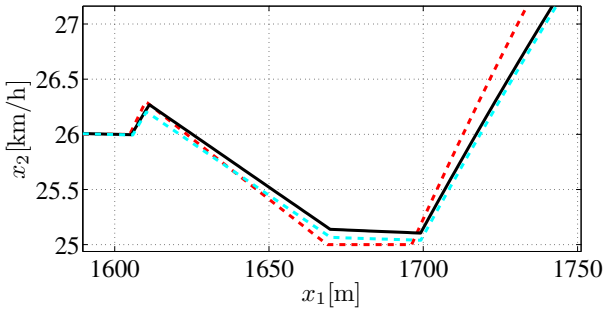


Figure 5.23: Detail of the closed-loop tracking response of Fig. 5.22.

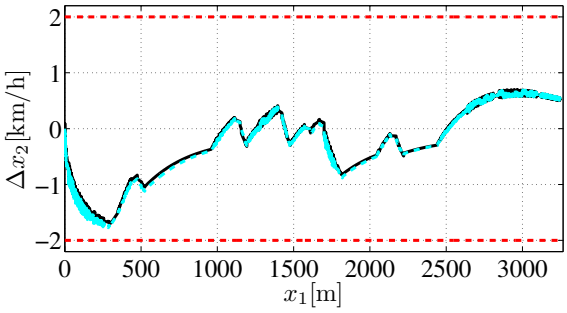


Figure 5.24: Closed-loop tracking error $\Delta x_2(k)$ with a mass variation of 20%. Red dashed lines: $\Delta x_{2min}(k)$ and $\Delta x_{2max}(k)$. Solid black line: tracking error $\Delta x_2(k)$ of the on-line solution of the problem (5.45) at each execution time. Cyan dashed line: tracking error $\Delta x_2(k)$ of Algorithm 4.

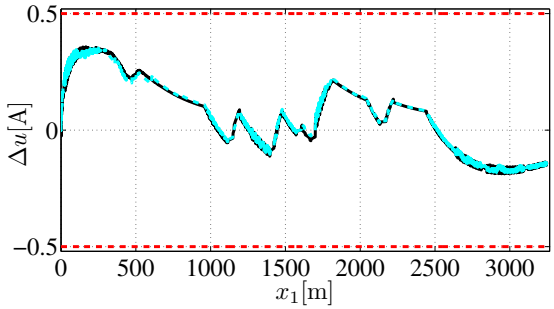


Figure 5.25: Closed-loop tracking error $\Delta u(k)$ with a mass variation of 20%. Red dashed lines: $\Delta u_{min}(k)$ and $\Delta u_{max}(k)$. Solid black line: tracking input $\Delta u(k)$ of the on-line solution of the problem (5.45) at each execution time. Cyan dashed line: tracking input $\Delta u(k)$ of Algorithm 4.

5.7.3 Comparison of fully on-line MPC based on PDLF and explicit MPC based on PDLF

The computational resources (computation time and memory) used by each approach on-line are reported in Table 5.2. In Table 5.2, also are recalled the results previously obtained in Table 5.1.

Table 5.2: On-line computational resources of the robust MPC for LPV systems and the off-line approach.

Method	**Computation time**	**Memory used**
Fully on-line computation of problem (5.16) in Section 5.3	21.4ms	4768Bytes
Algorithm 3: explicit MPC based on invariant ellipsoids	0.52ms	824Bytes
Fully on-line computation of problem (5.45) in Section 5.5	93.1ms	9928Bytes
Algorithm 4: Proposed explicit MPC using the PDLF and based on invariant ellipsoids	0.61ms	1968Bytes

From Table 5.2, and the results obtained previously, several facts can be highlighted:

- The approach (5.45), which includes the PDLF, requires longer computation time and larger memory resources that the approach (5.16), as expected,

since the quantity of LMIs involved in (5.45) is higher.

- The approach (5.45) has a better performance in terms of speed of convergence to the origin and smaller values for the performance cost γ, than the strategy (5.16), since the approach (5.45) is less conservative. This is illustrated in Fig. 5.26.

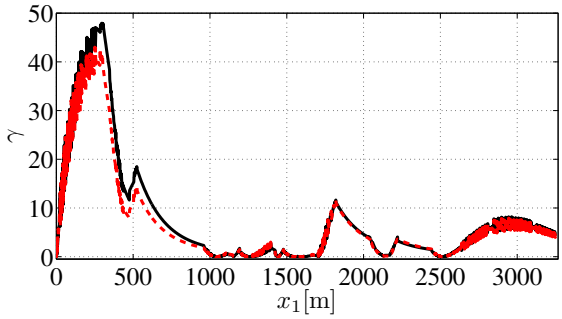


Figure 5.26: Performance cost $\gamma(k)$. Solid black line: MPC (5.16)[42]. Red dashed line: MPC with PDLF (5.45)[66].

- Algorithm 4 requires less computational time and memory resources than the strategy (5.45), as expected, since there are no LMIs involved in the on-line solution of Algorithm 4.

- Algorithm 4 and the strategy (5.45) have very similar performances, as was shown in Fig. 5.22-5.25. Therefore, the performance of Algorithm 4 is expected to be better than the performance of the strategy (5.16), and thus of Algorithm 3, since Algorithm 4 does include the parameter $\lambda(k)$ explicitly in the control law.

- Algorithm 4 requires larger memory resources than Algorithm 3. However, Algorithm 4 is still well suited to real-time implementation.

In the following section, Algorithm 4 will be embedded in a real-time device, and its performances will be tested for the *Vir'volt* vehicle in the benchmark (see Section 2.6).

5.8 Explicit MPC for LPV using PDLF: application to the benchmark

In this section, Algorithm 4 proposed in Section 5.6, is applied to the *Vir'volt* vehicle within the benchmark described in Section 2.6. Recall that the dynamics of the vehicle in the benchmark is a scaled version of the dynamics (2.6) and (2.7), and it is given by (2.48) in Section 2.6, i.e.

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k), \\ x_2(k+1) &= x_2(k) + 1.1228 T_s I_{batt}(k) - 0.1125 T_s x_2(k)^2 - 0.1893 T_s, \end{aligned} \tag{5.78}$$

where $u(k) = I_{batt}(k)$. The vehicle in the benchmark is expected to track the optimal driving strategy depicted in Fig. 2.14. The tracking error $\Delta x(k)$ is represented by the LPV model (5.1) and (5.2). The tracking constraints imposed to the problem (5.45), are

$$-3\text{km/h} \leq \Delta y(k) \leq 3\text{km/h and } -0.5\text{A} \leq \Delta u(k) \leq 0.5\text{A}. \tag{5.79}$$

The weighting matrices are $\mathbf{Q} = [0\ 0; 0\ 20]^T$ and $\mathbf{R} = 1$. The sampling period is $T_s = 0.2$s.

Algorithm 4 is embedded in the microcontroller dsPIC33ep512mu810 of Microchip®, described in Section 2.6. Algorithm 4 is applied to the vehicle in the benchmark as is described in the following.

- Off-line steps

  1. The gridding over the state Δx is chosen by making $\Delta x_1 = 0$ and

$$\begin{aligned} \Delta x_2 = [&4.17\text{km/h},\ 3.60\text{km/h},\ 2.27\text{km/h},\ 0.36\text{km/h},\\ &0.22\text{km/h},\ 0.09\text{km/h},\ 0.03\text{km/h}]. \end{aligned} \tag{5.80}$$

  2. For each element in (5.80), i.e. $\Delta x_2^{(i)}, i = 1, ..., 7$, the problem (5.45) is solved off-line, and the resulting $\mathbf{\Theta}_1^{(i)}$, $\mathbf{\Theta}_2^{(i)}$, $\mathbf{G}_1^{(i)}$, $\mathbf{G}_2^{(i)}$, $\mathbf{\Lambda}_1^{(i)}$, $\mathbf{\Lambda}_2^{(i)}$, $i = 1, ..., 7$, are saved in the memory of the dsPIC.

Each $\mathbf{\Theta}_1^{(i)}$ and $\mathbf{\Theta}_2^{(i)}$, $i = 1, ..., 7$, define the ellipsoids $\xi_1^{(i)}$ and $\xi_2^{(i)}$, given by (5.59) and (5.60), respectively. Those ellipsoids will be used on-line to find the corresponding invariant set $\Omega_1^{(i)}$ given by (5.61). In Fig. 5.27 and 5.28, are depicted the the ellipsoids $\xi_1^{(i)}$ and $\xi_2^{(i)}$ such that $\Omega_1^{(i)} = \xi_1^{(i)} \bigcap \xi_2^{(i)}$, $i = 1, ..., 7$.

- On-line steps

  1.-3. At each time k, the smallest invariant set $\Omega_1^{(i)}$ that contains the actual state $\Delta x(k)$, is found and the corresponding gains $\mathbf{\Lambda}_1^{(i)}$, $\mathbf{\Lambda}_2^{(i)}$, $\mathbf{G}_1^{(i)}$ and $\mathbf{G}_2^{(i)}$ are selected.

  4.-5. The control law

$$\Delta u(k) = \left(f_1(\lambda(k)) \mathbf{\Lambda}_1^{(i)} \left(\mathbf{G}_1^{(i)} \right)^{-1} + f_2(\lambda(k)) \mathbf{\Lambda}_2^{(i)} \left(\mathbf{G}_2^{(i)} \right)^{-1} \right) \Delta x(k), \tag{5.81}$$

  is computed and $u(k) = \Delta u(k) + u^*(k)$ is applied to the *Vir'volt* vehicle (see Fig. 2.11 with $I_{batt}(k) = u(k)$).

The computational resources (computation time and memory) used by Algorithm 4, embedded in the dsPIC, are reported in Table 5.3.

Table 5.3: Computational resources of Algorithm 4 embedded in the dsPIC.

ROM	**RAM**	**Loop time**
18%	2%	20ms

The tracking response of the explicit approach is depicted in Fig. 5.29. The tracking error $\Delta x_2(k)$, is depicted in Fig. 5.30. In Fig. 5.31, the control $\Delta u(k)$, in (5.81), is depicted. The control law $I_{batt}(k) = \Delta u(k) + u^*(k)$ applied to the *Vir'volt* vehicle is depicted in Fig. 5.32. Additionally, in Fig. 5.33, the open circuit voltage V_{oc} of the battery is depicted.

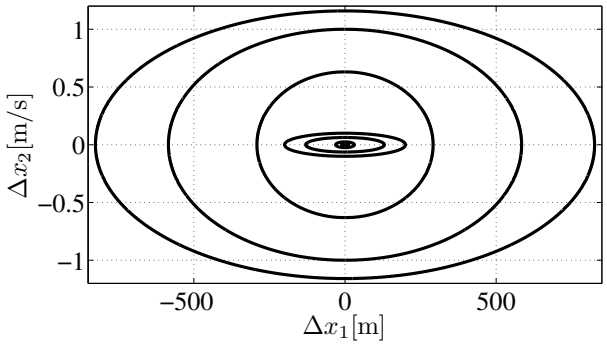


Figure 5.27: Ellipsoids $\xi_1^{(i)}$, $i = 1, ..., 7$.

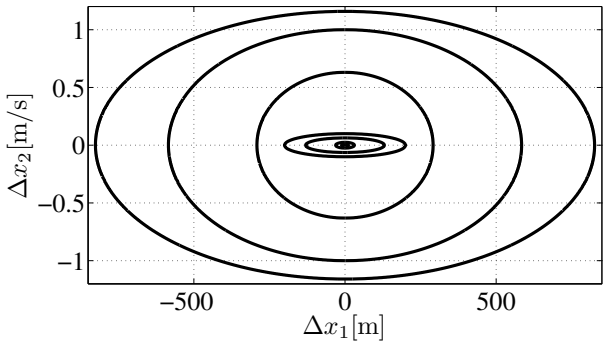


Figure 5.28: Ellipsoids $\xi_2^{(i)}$, $i = 1, ..., 7$.

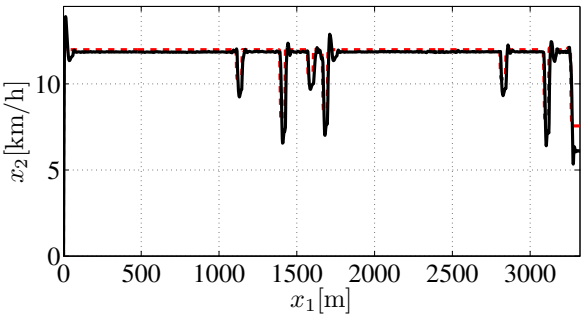


Figure 5.29: Closed-loop tracking response in the benchmark. Red dashed line: reference $x_2^*(k)$. Black solid line: measure of $x_2(k)$.

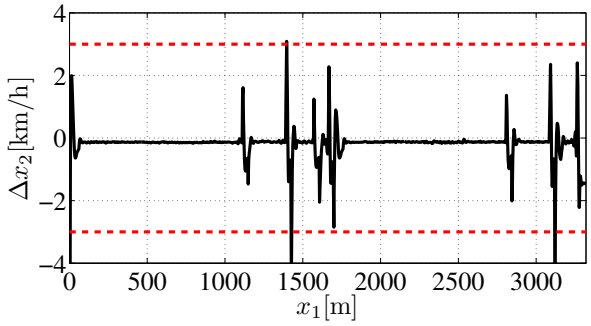


Figure 5.30: Tracking error $\Delta x_2(k)$ in the benchmark. Red dashed lines: $\Delta x_{2min}(k)$ and $\Delta x_{2max}(k)$. Solid black line: tracking error $\Delta x_2(k)$ obtained from the measure of $x_2(k)$.

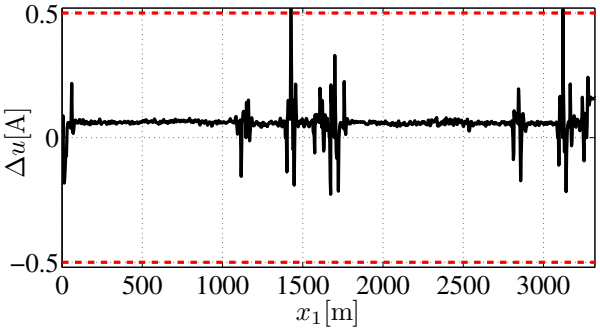


Figure 5.31: Control law $\Delta u(k)$ of the explicit algorithm. Red dashed lines: $\Delta u_{min}(k)$ and $\Delta u_{max}(k)$. Solid black line: control law $\Delta u(k)$.

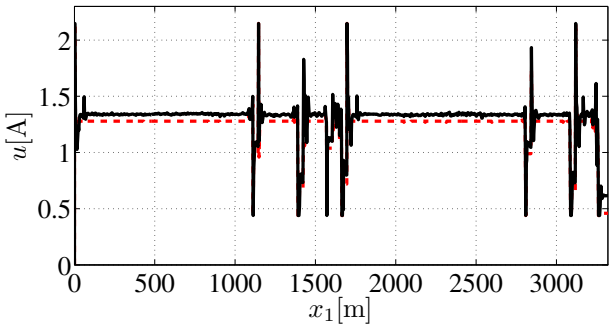


Figure 5.32: Input $u(k)$ to the Vir'volt vehicle in the benchmark ($I_{batt}(k) = u(k)$). Red dashed lines: expected value $u^*(k)$. Solid black line: input $u(k)$.

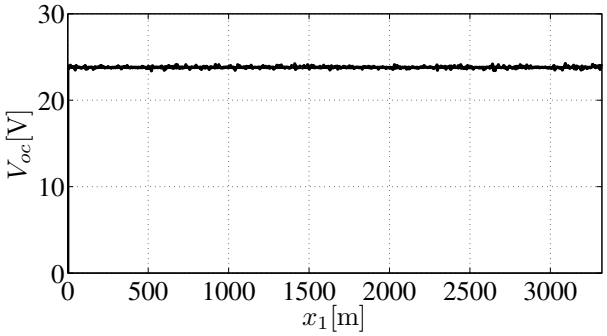


Figure 5.33: Open circuit voltage V_{oc} of the battery.

The results reported in Table 5.3, correspond to the memory required to save the look-up tables of the sets $\mathbf{\Theta}_1^{(i)}$, $\mathbf{\Theta}_2^{(i)}$, $\mathbf{G}_1^{(i)}$, $\mathbf{G}_2^{(i)}$, $\mathbf{\Lambda}_1^{(i)}$, $\mathbf{\Lambda}_2^{(i)}$, $i = 1, ..., 7$. Besides, the optimal reference $(x_1^*(t), x_2^*(t), I_{batt}^*(t))$, $t \in [0, t_f]$, was also saved in the ROM, in order to perform the tracking on-line.

From Fig. 5.29-5.32, it can be observed that, despite some peaks of short duration in $\Delta x_2(k)$, the performance of the explicit Algorithm 4 is well suitable to perform the tracking of the optimal driving strategy. The tracking is achieved despite the internal frictions in the benchmark, the physical vibrations between the vehicle and the inertia cylinder and the noise in the measurements. In Fig. 5.33, it can be observed that the open circuit voltage V_{oc} of the battery, remains constant during all the tracking task, therefore the initial hypothesis made in (2.10) in Section 2.3.1, of V_{oc} constant, is well verified. Hence, the solution of the optimization problem (2.15) corresponds to the minimization of the energetic consumption.

For illustrative purposes, consider now the same explicit Algorithm 4, carry out for the *Vir'volt* vehicle in the benchmark, as above, but with tracking constraints as

$$-3\text{km/h} \leq \Delta y(k) \leq 3\text{km/h} \text{ and } -2\text{A} \leq \Delta u(k) \leq 2\text{A}, \tag{5.82}$$

and the weighting matrix as $\mathbf{Q} = [0\ 0; 0\ 100]^T$. The rest is kept the same. Notice that the gain $\mathbf{Q}$ is bigger than the one considered above, therefore a more aggressive response is expected.

Algorithm 4 is performed using the same gridding in the state space as above, and finding a new set of ellipsoids and saving them in memory of the dsPIC. The tracking response of the explicit approach with $\mathbf{Q} = [0\ 0; 0\ 100]^T$, is depicted in Fig. 5.34. In Fig. 5.35, is depicted the tracking error $\Delta x_2(k)$. In Fig. 5.36, the control $\Delta u(k)$ is depicted. Finally, $I_{batt}(k)$ is depicted in Fig. 5.37.

Notice that from Fig. 5.31 and Fig. 5.36, the control law is more aggressive in 5.36 with $\mathbf{Q} = [0\ 0; 0\ 100]^T$, as expected. And therefore, the tracking error in 5.35 with $\mathbf{Q} = [0\ 0; 0\ 100]^T$, is in general smaller than the one in 5.30 with $\mathbf{Q} = [0\ 0; 0\ 20]^T$.

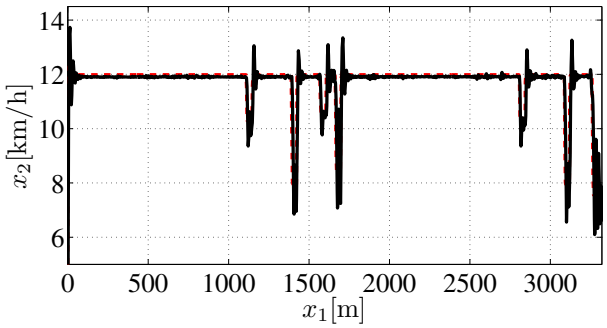


Figure 5.34: Closed-loop tracking response in the race benchmark. Red dashed line: reference $x_2^*(k)$. Black solid line: measure of $x_2(k)$.

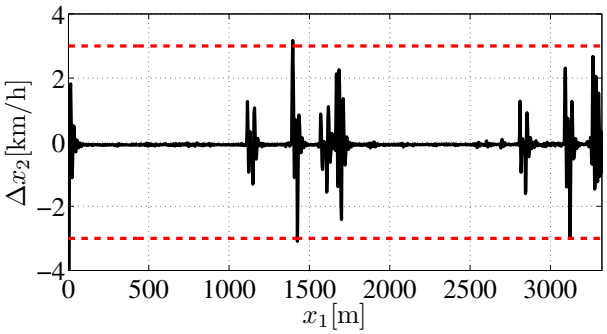


Figure 5.35: Tracking error $\Delta x_2(k)$ in the benchmark. Red dashed lines: $\Delta x_{2min}(k)$ and $\Delta x_{2max}(k)$. Solid black line: tracking error $\Delta x_2(k)$ obtained from the measure of $x_2(k)$.

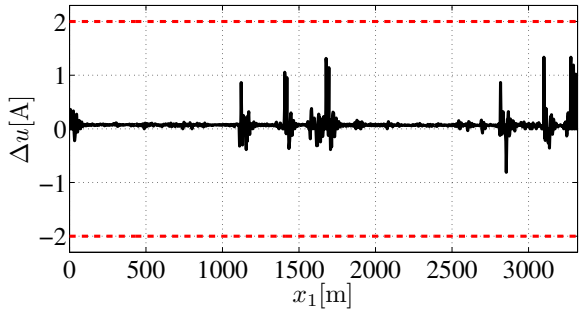


Figure 5.36: Control law $\Delta u(k)$ of the explicit algorithm. Red dashed lines: $\Delta u_{min}(k)$ and $\Delta u_{max}(k)$. Solid black line: control law $\Delta u(k)$.

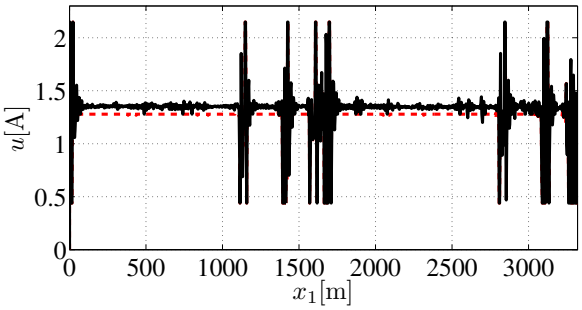


Figure 5.37: Input $u(k)$ to the Vir'volt vehicle in the benchmark ($I_{batt}(k) = u(k)$). Red dashed lines: expected value $u^*(k)$. Solid black line: input $u(k)$.

5.9 Conclusions

The tracking of the optimal driving strategy has been addressed through a LPV-based MPC approach. The dynamics of the tracking error was given by a LPV representation. The LPV framework has been motivated by two reasons: it preserves the nonlinear features of the dynamics, and the synthesis of the control law can be performed using efficient tools such as LMIs. An off-line strategy well suited for real-time implementation has been proposed. It reduces the computation times and the memory requirements needed to compute the parameter dependent control law, in comparison with existing approaches like [66]. It was shown that the resulting off-line approach is efficient for the present application. The explicit approach was embedded in a microcontroller and tested on-board in the benchmark.

Chapter 6

Robust adaptive real-time control based on an on-off driving srategy

6.1 Introduction

In this chapter, a complementary scheme of driving strategy, with respect to those presented in the previous part of this manuscript, is described. This scheme treats cases that were not considered before such as a time-varying and unknown dynamics, important traffic jam in the track and even a non constant efficiency of the power converter. However, similarly to the previous control schema, a main concern is still the compatibility with real-time implementation.

Recall the dynamics of the *Vir'volt* vehicle given by (2.6) and (2.7), i.e.

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t), \\ \frac{dx_2(t)}{dt} &= \frac{\eta k_t g_r}{m r_w} I_{batt}(t) - \frac{1}{2m}\rho C_d A_f x_2(t)^2 - g C_r \cos(\theta(t)) - g \sin(\theta(t)), \end{aligned} \tag{6.1}$$

where the parameters m, η, k_t, g_r, r_w, ρ, C_d, A_f, g and C_r, if unknown, have been identified off-line from experimental data. They are given as in Table 2.1 in Section 2.2.2.

In the previous part of this manuscript, it has been assumed that the model

(6.1) of the vehicle dynamics remains unchanged all along the race, or that changes in the parameters are small enough. If so, then the tracking task can benefit from the nominal robustness of the MPC strategy to sufficiently small disturbances and uncertainties [63]. In that case, the nominal system dynamic was identified using a nonlinear grey-box identification strategy (see Section 2.2.2), and the optimal driving strategy was fully computed off-line (see Section 2.3).

Consider now, the case in which the aerodynamics resistance may vary in virtue of the intensity and the direction of the wind during the race, but the boundaries of the uncertainty introduced by this phenomenon are not known a-priori. Besides, consider the case in which the rolling resistance coefficient is not constant due to irregularities of the path, and the boundaries of the variation in the rolling resistance are also not known a-priori. If the uncertainties in the parameters involved in the dynamics were small enough and its boundaries were known, then robust controllers, such as those described in the previous sections are suitable to perform the tracking task. However, if the boundaries of the uncertainties are too large, an amenable solution to handle this problem is the use of an adaptive control, in which the control adapts itself according to the actual value of the parameters involved in the dynamics.

Notice that the optimal driving strategy described in Section 2.3, which has been fully computed off-line, considers that the presence of other vehicles in the track does not require excessive braking or that they can be easily surpassed. This is in practice true most of the time, because during the race, the other vehicles in the track run slowly and there are just a few running at the same time. However, if the traffic jam is important (as it happens near the finish line and the pit lane entrance), it imposes strong breaking that may affect the validity of the optimal driving strategy computed off-line. This case is precisely the one considered in this chapter. Therefore, it might be necessary to recompute the driving strategy on-line to achieve the minimal consumption.

On the other hand, consider the case when the efficiency of the power converter may be better when working at full regime. Then the vehicle might be driven using an on-off strategy in order to reduce energetic losses. This case was not treated in previous part of this manuscript, where the efficiency of the power converter was constant (or considered as almost constant) for the range of

velocities involved in the driving strategy.

In this chapter, an adaptive robust control for the *Vir'volt* vehicle is presented. This adaptive driving strategy consists in an on-line identification of the dynamics (6.1) and a recomputation on-line of the on-off driving strategy that must achieve the minimal consumption.

The main features of the approach here proposed are:

- Very low computational burden, implementable on low-cost embedded micro-controller for real-time automatic control.
- Very low power consumption (less than 10 mW in average, about 15 J for the race).
- Real time adaptability with respect to changes in dynamics, weather or traffic conditions.

This chapter is organized as follows. In Section 6.2, the dynamics of the vehicle is represented with three coefficients, and the procedure of the identification of those parameters is described. In Section 6.3, an adaptive driving strategy is presented. Its main feature is a variable time horizon (the horizon used for optimization is deduced from the characteristics of the dynamics). It is shown that the optimal driving strategies exhibit a very simple periodic behaviour. Experimental results are presented in Section 6.4. This results were obtained during the Shell Eco-Marathon Europe 2014. Finally, in Section 6.5, conclusions are presented.

6.2 Parameter identification

Consider the dynamics of the *Vir'volt* vehicle (6.1). To be more general, the expression $\frac{dx_2(t)}{dt}$, in (6.1), is written as an ordinary differential equation of first degree. Then, (6.1) can be rewritten as follows

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t), \\ \frac{dx_2(t)}{dt} &= ax_2(t)^2 + bx_2(t) + c + \frac{\eta k_t g_r}{m r_w} I_{batt}(t), \end{aligned} \tag{6.2}$$

where a[1/m], b[1/s] and c[m/s^2] are unknown real quantities that need to be identified. At time $t \in [0, t_{max}]$, the vehicle is in position $x_1(t) \in [0, x_{1_{total}}]$ (t_{max} is the total duration of the race and $x_{1_{total}}$ is the total length of the race). Hereafter, it is assumed that $a < 0$, which means that the aerodynamic drag opposes to the movement of the vehicle (see 2.1).

Since the efficiency in the power converter is considered to be better when working at full regime, an on-off strategy is proposed. If the vehicle is driven using an on-off strategy, the expression (6.2) becomes

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t), \\ \frac{dx_2(t)}{dt} &= ax_2(t)^2 + bx_2(t) + c + \frac{\eta k_t g_r}{m r_w} I_{batt_{max}} u(t), \end{aligned} \tag{6.3}$$

where $I_{batt_{max}}$[A] is the maximum current of the battery, and the control $u(t) \in \{0, 1\}$ is a piecewise constant function accounting for the state of the motor. If the motor is off then $u = 0$ and if the motor is on then $u = 1$.

Since in practice, as will be described later, the deceleration phases are much more longer than the acceleration phases (the acceleration phases may be too short to acquire enough data to for proper identification), consider in (6.3) only the deceleration of the vehicle ($u = 0$). Therefore, the parameters to be identified are just a, b and c. The deceleration dynamics reads

$$\frac{dx_2(t)}{dt} = ax_2(t)^2 + bx_2(t) + c. \tag{6.4}$$

In the following, it is shown that in the case when the velocity of the wind during the race is not negligible, and in the case when the rolling resistance coefficient may vary due to irregularities of the path, then the parameters b and c may vary in time.

6.2.1 Problem formulation

Consider the forces reported on the scheme of Fig. 2.2, in Section 2.2.1. From (2.1), in Section 2.2.1, the vehicle is mainly subject to three kind of forces during

the deceleration, i.e.

$$m\frac{dx_2(t)}{dt} = -F_{aerodynamics}(t) - F_{rolling}(t) - F_g(t). \tag{6.5}$$

This forces are described as follows

- Aerodynamic force $F_{aerodynamics}(t)$: Consider the vector $\overrightarrow{\mathcal{V}_w}(t)$, as the vector of the velocity of the wind at time t. Consider also, the vector $\overrightarrow{x_2}(t)$ of the velocity of the vehicle, at time t. The component of $\overrightarrow{\mathcal{V}_w}(t)$ that is collinear with $\overrightarrow{x_2}(t)$, at time t, is denoted as $\overrightarrow{v}(t)$ and is given by the dot product of $\overrightarrow{\mathcal{V}_w}(t)$ and $\overrightarrow{x_2}(t)$. Due to the velocity of the wind $\overrightarrow{\mathcal{V}_w}(t)$, the vehicle undergoes a deceleration of the form

  $$a_1(x_2(t) - v(t))^2 + b_1(x_2(t) - v(t)), \tag{6.6}$$

  with $v(t) = |\overrightarrow{v}(t)|$, and $x_2(t) - v(t)$ the total velocity of the vehicle, at time t.

- Rolling resistance $F_{rolling}(t)$: The rolling resistance depends on the rolling friction coefficient, the weight of the vehicle and the slope θ[rad] of the track, assumed to be positive if the vehicle is going uphill and negative if it is going downhill, at time t. If the rolling friction coefficient varies in time, due for example to imperfections on the road, then the deceleration due to the rolling resistance reads

  $$c_4(t)\cos\theta(t). \tag{6.7}$$

  The slope of the road is assumed to be known a-priori according to the position of the vehicle in the track, but in general it may be unknown.

- Gravitational force $F_g(t)$: The gravitational force depends on the slope of the track at time t and on the weight of the vehicle. Due to the gravitational force, the vehicle undergoes a deceleration (or acceleration according to the sign of θ[rad]) of the form

  $$c_3\sin\theta(t). \tag{6.8}$$

Plugging (6.6), (6.8), and (6.7) into (6.5), gives

$$\frac{dx_2(t)}{dt} = -a_1(x_2(t) - v(t))^2 - b_1(x_2(t) - v(t)) - c_3 \sin\theta(t) - c_4(t)\cos\theta(t). \quad (6.9)$$

Since, in practice, the velocity of the wind is not larger than the velocity of the vehicle, i.e. $x_2(t) - v(t) > 0$, for all t, then (6.9) can be rewritten as

$$\frac{dx_2(t)}{dt} = -a_1x_2(t)^2+(2a_1v(t)-b_1)x_2(t)-a_1v(t)^2+b_1v(t)-c_3\sin\theta(t)-c_4(t)\cos\theta(t). \quad (6.10)$$

Notice that neither (6.8) nor (6.7) depends on the vehicle velocity. Comparing (6.10) with respect to (6.4), it is obtained that

$$\begin{aligned} a &= -a_1, \\ b &= b(t) = 2a_1v(t) - b_1, \\ c &= c(t) = -a_1v(t)^2 + b_1v(t) - c_3\sin\theta(t) - c_4(t)\cos\theta(t). \end{aligned} \quad (6.11)$$

Therefore b and c may vary in time, if the wind during the race is not negligible and important irregularities in the path are considered. Notice that the parameter a is constant since it depends exclusively on the features of the vehicle, not on the conditions of the race. Equation (6.4) can be rewritten as

$$\frac{dx_2(t)}{dt} = ax_2(t)^2 + b(t)x_2(t) + c(t). \quad (6.12)$$

In the following subsections, a strategy to estimate the parameters a, $b(t)$ and $c(t)$ from the deceleration behaviour of the vehicle is presented. First, the time-invariant parameter a is identified off-line from experimental data. Next, the parameters $b(t)$ and $c(t)$ are identified on-line as they vary in time.

6.2.2 Identification of parameter a

In the following, the time-invariant parameter a is identified off-line. The identified parameter a will be used later to identify $b(t)$ and $c(t)$ on-line.

The key idea is to measure the velocities $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$, $x_2(t_3)$,..., at time instants t_0, t_1, t_2, t_3,..., during the deceleration of the vehicle, and to use

those measurements to find off-line the parameter a involved in (6.12). For the sake of simplicity, the measurements will be uniformly spaced in time, i.e. $t_1 - t_0 = t_2 - t_1 = t_3 - t_2 = \dots$. Hereafter, the following assumption holds.

Assumption 5 *The race conditions such as wind, traffic, road features, etc., are assumed to remain constant at least during four consecutive measurements of the velocity, i.e. from $x_2(t_0)$ to $x_2(t_3)$, and therefore $b(t_0) = b(t_1) = b(t_2) = b(t_3)$ and $c(t_0) = c(t_1) = c(t_2) = c(t_3)$.*

Consider three velocities $x_2(t_0)$, $x_2(t_1)$ and $x_2(t_2)$ which are measured during the deceleration of the vehicle. It can be shown (see Appendix C.1) that from the solution of the differential equation (6.12), the following linear equation is obtained for $b(t_0) = b(t_1) = b(t_2) = b$ and $c(t_0) = c(t_1) = c(t_2) = c$

$$\begin{aligned}\left[x_2(t_1)^2(x_2(t_0) + x_2(t_2)) - 2x_2(t_0)x_2(t_1)x_2(t_2)\right] a \\ + \left(x_2(t_1)^2 + x_2(t_0)x_2(t_2)\right) b + \left[2x_2(t_1) - (x_2(t_0) + x_2(t_2))\right] c = 0.\end{aligned} \tag{6.13}$$

Notice that in (6.13), there are three unknown parameters a, b and c. Then, in principle, three equations are required to solve the system. If five velocities $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$, $x_2(t_3)$ and $x_2(t_4)$ are measured, the following system of three linear equations is obtained ($b(t_0) = b(t_1) = b(t_2) = b(t_3) = b(t_4) = b$ and $c(t_0) = c(t_1) = c(t_2) = c(t_3) = c(t_4) = c$)

$$\begin{aligned}\left[x_2(t_1)^2(x_2(t_0) + x_2(t_2)) - 2x_2(t_0)x_2(t_1)x_2(t_2)\right] a \\ + \left(x_2(t_1)^2 + x_2(t_0)x_2(t_2)\right) b + \left[2x_2(t_1) - (x_2(t_0) + x_2(t_2))\right] c = 0,\end{aligned} \tag{6.14}$$

$$\begin{aligned}\left[x_2(t_2)^2(x_2(t_1) + x_2(t_3)) - 2x_2(t_1)x_2(t_2)x_2(t_3)\right] a \\ + \left(x_2(t_2)^2 + x_2(t_1)x_2(t_3)\right) b + \left[2x_2(t_2) - (x_2(t_1) + x_2(t_3))\right] c = 0,\end{aligned} \tag{6.15}$$

$$\left[x_2(t_3)^2(x_2(t_2)+x_2(t_4))-2x_2(t_2)x_2(t_3)x_2(t_4)\right]a + \left(x_2(t_3)^2+x_2(t_2)x_2(t_4)\right)b+\left[2x_2(t_3)-(x_2(t_2)+x_2(t_4))\right]c=0. \tag{6.16}$$

However, this system of equations does not admit an unique solution (the system is not linearly independent), but an affine solution in which one quantity a, b or c is imposed and the other ones are found. Therefore, only four measured velocities, $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$ and $x_2(t_3)$, and only two equations, i.e. (6.14) and (6.15), are needed to estimate the affine solutions a, b and c.

On the other hand, notice that the equation (6.12) does have an unique solution. And yet, if the imposed parameter (a, b or c) is not properly chosen, the affine solution of the homogeneous linear equations (6.14) and (6.15) may not be a solution of (6.12).

In the following, the measurements of $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$ and $x_2(t_3)$, and the systems of equations (6.14) and (6.15), are used to estimate the values a, b and c that are also solution of (6.12), under Assumption 5. This will give the parameter a that will remain constant during the race, since it does not depend of the race conditions (see (6.11)).

6.2.2.1 Algorithm of the off-line identification

Consider the measurements $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$, $x_2(t_3)$. Consider also an initial non-zero value $a_{(1)} \in \mathbb{R}$. If a is fixed as $a = a_{(1)}$, then the linear system of equations (6.14) and (6.15) has an unique solution denoted as $b = b_{(1)}$ and $c = c_{(1)}$. For any other value $a_{(s)}$, such that $a_{(s)} = a_{(1)}s$, with $s \in \mathbb{R}$, then the unique solution of (6.14) and (6.15) is $b_{(s)} = b_{(1)}s$ and $c_{(s)} = c_{(1)}s$.

The objective is to find the value s such that the solution $a_{(s)}$, $b_{(s)}$ and $c_{(s)}$, is also solution of (6.12), with $a_{(1)}$, $b_{(1)}$ and $c_{(1)}$ known.

First, consider a fixed as $a_{(1)} = -1$, then the system of equations (6.14) and

(6.15), becomes

$$\begin{aligned}
&-\left[x_2(t_1)^2(x_2(t_0)+x_2(t_2))-2x_2(t_0)x_2(t_1)x_2(t_2)\right]\\
&\qquad+\left(x_2(t_1)^2+x_2(t_0)x_2(t_2)\right)b+\left[2x_2(t_1)-(x_2(t_0)+x_2(t_2))\right]c=0,\\
&-\left[x_2(t_2)^2(x_2(t_1)+x_2(t_3))-2x_2(t_1)x_2(t_2)x_2(t_3)\right]\\
&\qquad+\left(x_2(t_2)^2+x_2(t_1)x_2(t_3)\right)b+\left[2x_2(t_2)-(x_2(t_1)+x_2(t_3))\right]c=0.
\end{aligned} \tag{6.17}$$

The system of equations (6.17) has an unique solution given by

$$a_{(1)}=-1,\ b=b_{(1)} \text{ and } c=c_{(1)}. \tag{6.18}$$

However, $a_{(1)}$, $b_{(1)}$ and $c_{(1)}$ are not necessarily solution of (6.12).

Let suppose that there exists $s=s^*$, such that the affine solution $a_{(s^*)}$, $b_{(s^*)}$ and $c_{(s^*)}$ of (6.17), i.e.

$$\begin{aligned}
a_{(s^*)}&=a_{(1)}s^*=-s^*,\\
b_{(s^*)}&=b_{(1)}s^*,\\
c_{(s^*)}&=c_{(1)}s^*,
\end{aligned} \tag{6.19}$$

is also solution of (6.12) under Assumption 5.

It is possible to show (see (C.4) and (C.6) in Appendix C.1) that if $a_{(s^*)}$, $b_{(s^*)}$, $c_{(s^*)}$ are solution of (6.12), then $a_{(s^*)}$, $b_{(s^*)}$ and $c_{(s^*)}$ verify

$$\begin{aligned}
&\sqrt{{b_{(s^*)}}^2-4a_{(s^*)}c_{(s^*)}}\\
&=\frac{1}{t_1-t_0}\left(\log\left(\frac{2a_{(s^*)}x_2(t_1)-\sqrt{{b_{(s^*)}}^2-4a_{(s^*)}c_{(s^*)}}+b_{(s^*)}}{2a_{(s^*)}x_2(t_1)+\sqrt{{b_{(s^*)}}^2-4a_{(s^*)}c_{(s^*)}}+b_{(s^*)}}\right)\right.\\
&\qquad\left.-\log\left(\frac{2a_{(s^*)}x_2(t_0)-\sqrt{{b_{(s^*)}}^2-4a_{(s^*)}c_{(s^*)}}+b_{(s^*)}}{2a_{(s^*)}x_2(t_0)+\sqrt{{b_{(s^*)}}^2-4a_{(s^*)}c_{(s^*)}}+b_{(s^*)}}\right)\right),
\end{aligned} \tag{6.20}$$

where $x_2(t_0)$ and $x_2(t_1)$ are velocities measured at time t_0 and t_1. Plugging (6.19)

into (6.20), gives

$$
\begin{aligned}
&s^* \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} \\
&= \frac{1}{t_1 - t_0} \left(\log \left(\frac{2a_{(1)}x_2(t_1) - \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}}{2a_{(1)}x_2(t_1) + \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}} \right) \right. \\
&\qquad \left. - \log \left(\frac{2a_{(1)}x_2(t_0) - \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}}{2a_{(1)}x_2(t_0) + \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}} \right) \right),
\end{aligned}
\tag{6.21}
$$

from which

$$
\begin{aligned}
s^* = &\frac{1}{(t_1 - t_0)\sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}}} \left(\log \left(\frac{2a_{(1)}x_2(t_1) - \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}}{2a_{(1)}x_2(t_1) + \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}} \right) \right. \\
&\left. - \log \left(\frac{2a_{(1)}x_2(t_0) - \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}}{2a_{(1)}x_2(t_0) + \sqrt{{b_{(1)}}^2 - 4a_{(1)}c_{(1)}} + b_{(1)}} \right) \right).
\end{aligned}
\tag{6.22}
$$

Therefore, the solution $a_{(s^*)} = -s^*$ of (6.17), is also solution of (6.12) with s^* given by (6.22). The identified parameter $a_{(s^*)}$ is kept, since its describes the features of the vehicle. The parameters b and c need to be identified on-line since they depend on the conditions of the race (see 6.11).

This is valid for any t_0 and any t_1 such that $t_0 < t_1$. In practice, to improve the precision, it is important to avoid involving factors too small in the solution (6.22) of s^*, and it will be preferable to make $t_1 - t_0$ not too small, so $x_2(t_1)$ and $x_2(t_0)$ will be different enough (under Assumption 5).

6.2.3 Identification of parameters b and c

Once the parameter $a = -s^*$ has been identified off-line, the parameters $b(t)$ and $c(t)$ in (6.12) can be identified on-line under Assumption 5.

Consider the on-line measurements $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$, $x_2(t_3)$, and the sys-

tem of linear equations (6.17) with $a = a_{(s^*)} = -s^*$, i.e.

$$\begin{aligned}
&-\left[x_2(t_1)^2(x_2(t_0)+x_2(t_2)) - 2x_2(t_0)x_2(t_1)x_2(t_2)\right] s^* \\
&\qquad + \left(x_2(t_1)^2 + x_2(t_0)x_2(t_2)\right) b(t) + \left[2x_2(t_1) - (x_2(t_0)+x_2(t_2))\right] c(t) = 0, \\
&-\left[x_2(t_2)^2(x_2(t_1)+x_2(t_3)) - 2x_2(t_1)x_2(t_2)x_2(t_3)\right] s^* \\
&\qquad + \left(x_2(t_2)^2 + x_2(t_1)x_2(t_3)\right) b(t) + \left[2x_2(t_2) - (x_2(t_1)+x_2(t_3))\right] c(t) = 0,
\end{aligned} \tag{6.23}$$

with $b(t)$ and $c(t)$ assumed constant for $t \in [t_0, t_3]$ (see Assumption 5). Introducing the auxiliary variables

$$\begin{aligned}
\hat{\alpha}_1 &= \left[x_2(t_1)^2(x_2(t_0)+x_2(t_2)) - 2x_2(t_0)x_2(t_1)x_2(t_2)\right] s^*, \\
\hat{\beta}_1 &= \left(x_2(t_1)^2 + x_2(t_0)x_2(t_2)\right), \\
\hat{\gamma}_1 &= \left[2x_2(t_1) - (x_2(t_0)+x_2(t_2))\right],
\end{aligned} \tag{6.24}$$

and

$$\begin{aligned}
\hat{\alpha}_2 &= \left[x_2(t_2)^2(x_2(t_1)+x_2(t_3)) - 2x_2(t_1)x_2(t_2)x_2(t_3)\right] s^*, \\
\hat{\beta}_2 &= \left(x_2(t_2)^2 + x_2(t_1)x_2(t_3)\right), \\
\hat{\gamma}_2 &= \left[2x_2(t_2) - (x_2(t_1)+x_2(t_3))\right],
\end{aligned} \tag{6.25}$$

the system of equations (6.23) can be rewritten as

$$\begin{aligned}
-\hat{\alpha}_1 + \hat{\beta}_1 b(t) + \hat{\gamma}_1 c(t) = 0, \\
-\hat{\alpha}_2 + \hat{\beta}_2 b(t) + \hat{\gamma}_2 c(t) = 0.
\end{aligned} \tag{6.26}$$

The solution of (6.26), and thus of (6.23), is

$$\begin{aligned}
b(t) &= \frac{-\hat{\alpha}_1\hat{\gamma}_2 + \hat{\alpha}_2\hat{\gamma}_1}{\hat{\alpha}_1\hat{\beta}_2 - \hat{\beta}_1\hat{\gamma}_2}, \\
c(t) &= \frac{(\hat{\alpha}_1)^2\hat{\beta}_2 - \hat{\alpha}_1\hat{\beta}_1\hat{\gamma}_2 + \hat{\alpha}_1\hat{\gamma}_2 - \hat{\alpha}_2\hat{\gamma}_1}{\hat{\alpha}_1\hat{\beta}_2\hat{\gamma}_1 - \hat{\beta}_1\hat{\gamma}_1\hat{\gamma}_2}.
\end{aligned} \tag{6.27}$$

The algorithm that identifies on-line the parameters $b(t)$ and $c(t)$ involved in (6.12), is summed up in the following.

Algorithm 5

- Off-line steps
  1. From the measurements $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$, $x_2(t_3)$ performed during the deceleration, solve (6.17) and find $b_{(1)}$ and $c_{(1)}$.
  2. Using $a_{(1)} = -1$, $b_{(1)}$ and $c_{(1)}$, solve (6.22) and find s^*.
  3. Make $a = -s^*$.
- On-line steps
  1. During the deceleration, measure $x_2(t_0)$, $x_2(t_1)$, $x_2(t_2)$, $x_2(t_3)$ at time t_0, t_1, t_2 and t_3.
  2. Using (6.24), (6.25) and (6.27), find $b(t)$ and $c(t)$.
  3. Return to Step 1.

Once the dynamics has been identified on-line, a low consumption driving strategy can be derived. This driving strategy will depend on the identified parameters a, $b(t)$ and $c(t)$. The driving strategy must guarantee the lowest energetic consumption.

6.3 Low consumption driving strategy

6.3.1 Problem formulation

Consider the dynamics (6.3), i.e.

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t), \\ \frac{dx_2(t)}{dt} &= ax_2(t)^2 + b(t)x_2(t) + c(t) + \frac{\eta k_t g_r}{r_w} I_{batt_{max}} u(t), \end{aligned} \tag{6.28}$$

where the parameter a has been identified off-line, and the parameters $b(t)$ and $c(t)$ are identified on-line. Since $u(t) \in \{0, 1\}$ is a piecewise constant function, the dynamics (6.28) is considered as switching system.

In an on-off driving strategy, when the motor is off ($u(t) = 0$), the energy consumption of the vehicle is reduced to a roughly constant residual consumption of the electronics (a few milliwatts). When the motor is on ($u(t) = 1$), the energy consumption of the vehicle goes up to 150W. For about 70% of the time, the motor is off. In this case, the transmission parts (gears, chains,...) are decoupled from the wheels and rapidly stop (the kinetic energy is dissipated by the friction and the Joule effect in the motor). When the motor is switched on, the transmission parts (gears, chains, ..) that were at rest, have to speed up in order to reach the wheels velocity, before the coupling allows to the energy to be transmitted from the motor to the wheels. The energetic cost of switching on the motor will be denoted as C_{on}, and it is about 10J.

For every admissible control $u(t)$ in (6.28), the following energy cost can be associated

$$C(u(t)) = \int_0^{t_{f_{max}}} h(x_2(t), u(t))\mathrm{d}t + C_{on}\, ND(u(t)), \tag{6.29}$$

where $t_{f_{max}}$[s] is the maximum time allowed to complete the race, $h : [0, +\infty) \times \{0, 1\} \to [0, +\infty)$ is a C^1 function, $C_{on} \geq 0 \in \mathbb{R}$ is the switching cost and $ND(u(t))$ denotes the number of discontinuities in $u(t)$.

Optimal control problem 2

The objective is to find a piecewise constant control $u(t)$ that steers the position of the vehicle $x_1(t)$, from its initial position $x_1(0) = 0$, to the final position $x_1(t_f) = x_{1_{total}}$, where $x_{1_{total}}$[m] is the total distance to run, in a time t_f such that $t_f \leq t_{f_{max}}$, with $t_{f_{max}}$[s] the maximum time allowed to complete the race. The piecewise constant control $u(t)$ must minimize the cost (6.29), subject to the dynamics (6.28) and with the velocity of the vehicle $x_2(t)$ subject to the following constraints

$$\begin{aligned} &x_2(t) \leq x_{2max}, \text{ if the vehicle is in a straight line,} \\ &x_2(t) \leq x_{2_{curve_{max}}} \leq x_{2max}, \text{ if the vehicle is in a curve,} \end{aligned} \tag{6.30}$$

where $x_{2max}$is the maximum velocity of the vehicle, and $x_{2_{curve_{max}}}$ is the maximum velocity allowed in a curve to avoid the vehicle slipping over (for further details refer to Section 2.3.2.1).

Regarding the solution of the Optimization problem 2, the following former results on the optimal control of switched systems can be addressed.

6.3.2 Former results

Switching systems are widely present in most of the industrial processes [55] and represent a major field of study in automatic control [20]. The optimal control of such switched systems is now pretty well understood for linear dynamics [59, 64, 37, 47].

For nonlinear dynamics, the case without switching cost, i.e. $C_{on} = 0$ in (6.29), has raised considerable attention, see for instance [35, 70, 14, 13, 44, 8]. When $C_{on} = 0$, the optimization problem is classically tackled with the Pontryaguin Maximum Principle (PMP), see for example [9] for an intrinsic formulation.

Non smooth extensions of the PMP have been also developed to include the case of non smooth trajectories or nonzero switching costs [65, 56].

Nevertheless, two main difficulties prevent the direct use of these results in the present application.

- The first difficulty arises when modelling the dynamics. In the standard frame of Lipschitz continuous dynamics, a vehicle with switched off motor on a flat road will never be at rest because of the principle of non-intersection of solutions. A more realistic modelling (where the vehicles can actually stop) requires to consider less regular dynamics, which usually implies to non-trivial technical subtleties. This explain that this approach is scarce in practice (with few exceptions, such as [43] for instance).

- The second difficulty is the heavy computational burden. Even in the standard Lipschitz continuous framework, with a specially adapted algorithm [38, 25, 26] and powerful computers, some dozen of seconds of computations

are needed with a powerful desktop computer to compute the optimal strategy of a 39 minutes run, and any perturbation (traffic, weather change,...) requires a new computation. Additionally, on-line integration of the PMP is not reasonable without a strong multi-core processor ($\approx$ 50W in average, about 100 kJ for the race). On the other hand, look-up tables with optimal trajectories may be pre-computed off-line and stored in memory, but such databases will be quite large: considering 1000 points for the position on the track, 300 points for the speed, 600 points for the time and 10 points for the wind speed and lead to about 2.10^9 entries. The construction of such a look-up table requires a good knowledge of the geometry of the track and the search in this base is time consumptive for low cost embedded devices with limited bandwidth.

Since global optimization are out of reach, in the following section, a robust control algorithm that solves on-line the Optimization problem 2, is presented.

6.3.3 Periodic low consumption driving strategy

The driving strategy here proposed will consist in letting the velocity $x_2(t)$ of the vehicle periodically oscillate between two values V_{min} and V_{max}. Those values will be find on-line from the dynamics of the vehicle, that is identified also on-line. V_{min} and V_{max} will lead to the lowest energetic consumption. As will be shown later, the computation of the periodic driving strategy will benefit of a low computational cost. Hereafter the following assumptions hold.

Assumption 6

1. The track and weather conditions are continuous.
2. For $u(t)$ constant in $\{0, 1\}$, and for every $x_1(t)$ and $x_2(t)$ in (6.28), the Cauchy problem (6.28) with initial condition $x_1(t_0)$ and $x_2(t_0)$, at time t_0, admits a locally unique solution in positive time.
3. The friction losses increase with speed.

4. The slope of the track and the wind action are never strong enough to prevent the car to go forward.

5. The car goes faster when motor is switched on.

6. When motor is off, the velocity of the vehicle tends to zero.

7. The energy consumption is zero when the motor is off, positive when motor is on.

8. The energy consumption increases with speed.

9. The switching cost is not so large, so the best strategy on long term is not necessarily to keep full speed.

Define the average velocity $\bar{V}$, in the time interval $t \in [t_0, t_1]$, as

$$\bar{V} = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} x_2(t) \mathrm{d}t, \tag{6.31}$$

with $x_2(t)$ absolutely continuous and $u(t)$ piecewise constant, and $x_2(t_0) = x_2(t_1)$ and $u(t_0) = u(t_1)$ (see Fig. 6.1). Notice that only trajectories for which the final velocity $x_2(t_1)$ is equal to the initial velocity $x_2(t_0)$ and the final value of the control $u(t_1)$ is equal to initial one $u(t_0)$, are considered. In particular, the control $u(t)$ has an even number of discontinuities in $t \in [t_0, t_1]$.

A piecewise constant function $u(t) \in \{0, 1\}$, with $t \in [t_0, t_1]$, is an *admissible control for the average speed* $\bar{V}$ if there exists a function $x_2(t) \in \mathbb{R}$, such that $\bar{V}$ is the average speed of $x_2(t)$ in $t \in [t_0, t_1]$.

Since the energy consumption is non-zero only when the motor is on, for an admissible control $u(t)$, with an average speed $\bar{V}$ in $t \in [t_0, t_1]$, the following cost can be defined

$$C(u(t))_{[t_0,t_1]} = \frac{C_{on} + V_{oc}(t) I_{batt_{max}} t_{on}}{t_1 - t_0}, \tag{6.32}$$

where C_{on} is the cost of turning on the motor, $V_{oc}(t)$ is the open circuit voltage of the battery (assumed to be not necessarily constant but accessible at each time t), $I_{batt_{max}}$ is the maximum current of the battery, and t_{on} is the effective

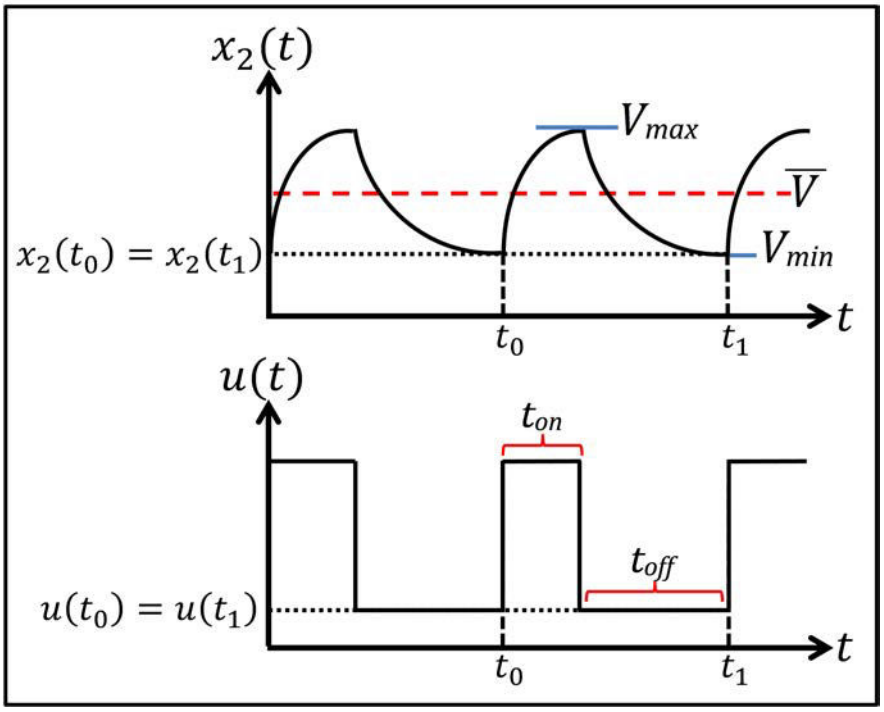


Figure 6.1: Periodic driving strategy.

time in which the motor is on in the time interval $[t_0, t_1]$ (see Fig. 6.1). The cost $C(u(t))_{[t_0,t_1]}$, in (6.32), is the average energy consumption used to run the distance $x_1(t_1) - x_1(t_0) = (t_1 - t_0)\bar{V}$ in time $t_1 - t_0$.

The driving strategy here proposed will consist in letting the velocity $x_2(t)$ periodically oscillate between $V_{min}(t)$ and $V_{max}(t)$ properly chosen such that the smallest $C(u(t))_{[t_0,t_1]}$ is obtained. The values $V_{min}(t)$ and $V_{max}(t)$ are computed on-line at time t. The values $V_{min}(t)$ and $V_{max}(t)$ must ensure that the race is finished in a time $t_f \leq t_{f_{max}}$. To this purpose, define the average speed $\bar{V}^*(t)$, required at time t to finish the race in a time $t_f \leq t_{f_{max}}$, as follows

$$\bar{V}^*(t) = \frac{x_{1_{total}} - x_1(t)}{t_{f_{max}} - t}, \tag{6.33}$$

where the position $x_1(t)$ of the vehicle is actually the travelled distance at time t.

The values $V_{min}(t)$ and $V_{max}(t)$ such that the average velocity $\bar{V}^*(t)$ (required at time t to finish the race on time) is obtained, read

$$\bar{V}^*(t) = \frac{V_{max}(t)t_{on} + V_{min}(t)t_{off}}{t_{on} + t_{off}}, \tag{6.34}$$

where t_{off} is the effective time in which the motor is off (see Fig. 6.1). The values $V_{max}(t)$, $V_{min}(t)$, t_{on} and t_{off}, are unknown. In the following, an on-line iterative algorithm to find $V_{max}(t)$ and $V_{min}(t)$ such that $\bar{V}^*(t)$ is obtained at time t, is described. This algorithm will use the dynamics that has been identified with Algorithm 5 in Section 6.2.3.

Algorithm 6

1. Measure $x_1(t)$ at time t and compute $\bar{V}^*(t)$ using (6.33).

2. If $\bar{V}^*(t) \geq \min\{x_{2_{curve_{max}}}, x_{2_{max}}\}$, according to the actual position $x_1(t)$ of the vehicle (see (6.30)), fix $V_{max}(t) = \min\{x_{2_{curve_{max}}}, x_{2_{max}}\}$ and $V_{min}(t) = V_{max}(t) - \delta$, where δ is a small enough constant ($\delta = 0.5$m/s), and go to Step 8. If not, if $\bar{V}^*(t) < \min\{x_{2_{curve_{max}}}, x_{2_{max}}\}$, choose arbitrarily a value $V_{min}(t)$ such as $V_{min}(t) < \bar{V}^*(t)$, and go to Step 3.

3. Choose arbitrarily a value $V_{max}(t)$ such that $V_{max}(t) > \bar{V}^*(t)$.

4. Use the identified parameters a, $b(t)$ and $c(t)$ (identified with Algorithm 5 at time t) to find t_{on} when the motor is on. The time t_{on} is the time that will take the system to go from $x_2(t_0) = V_{min}$ to $x_2(t_1) = V_{max}$, with $t_1 - t_0 = t_{on}$. From the solution of (6.28), with $u(t) = 1$, the time t_{on} is

given by (see Appendix C.1)

$$t_{on} = \frac{1}{\sqrt{b(t)^2 - 4a\left(c(t) + \frac{\eta k_t g_r}{r_w} I_{batt_{max}}\right)}} \left(\log\left(\frac{2aV_{max}(t) - \sqrt{b(t)^2 - 4a\left(c(t) + \frac{\eta k_t g_r}{r_w} I_{batt_{max}}\right)} + b(t)}{2aV_{max}(t) + \sqrt{b(t)^2 - 4a\left(c(t) + \frac{\eta k_t g_r}{r_w} I_{batt_{max}}\right)} + b(t)} \right) - \log\left(\frac{2aV_{min}(t) - \sqrt{b(t)^2 - 4a\left(c(t) + \frac{\eta k_t g_r}{r_w} I_{batt_{max}}\right)} + b(t)}{2aV_{min}(t) + \sqrt{b(t)^2 - 4a\left(c(t) + \frac{\eta k_t g_r}{r_w} I_{batt_{max}}\right)} + b(t)} \right) \right). \tag{6.35}$$

5. Use the identified parameters a, $b(t)$ and $c(t)$ to find t_{off}, that is the time that will take the system to go from $x_2(t_0) = V_{max}$ to $x_2(t_1) = V_{min}$, with $t_1 - t_0 = t_{off}$ and the motor off. From the solution of (6.28), with $u(t) = 0$, the time t_{off} is given by (see Appendix C.1)

$$t_{off} = \frac{1}{\sqrt{b(t)^2 - 4ac(t)}} \left(\log\left(\frac{2aV_{min}(t) - \sqrt{b(t)^2 - 4ac(t)} + b(t)}{2aV_{min}(t) + \sqrt{b(t)^2 - 4ac(t)} + b(t)} \right) - \log\left(\frac{2aV_{max}(t) - \sqrt{b(t)^2 - 4ac(t)} + b(t)}{2aV_{max}(t) + \sqrt{b(t)^2 - 4ac(t)} + b(t)} \right) \right). \tag{6.36}$$

6. Use $V_{max}(t)$, $V_{min}(t)$, t_{on} and t_{off} to solve

$$\bar{V}(t) = \frac{V_{max}(t)t_{on} + V_{min}(t)t_{off}}{t_{on} + t_{off}}. \tag{6.37}$$

7. If $\bar{V}(t) < \bar{V}^*(t)$, choose a bigger $V_{max}(t)$ and return to Step 1. If $\bar{V}(t) > \bar{V}^*(t)$, choose a smaller $V_{max}(t)$ and return to Step 1. Repeat using a search by dichotomy until $\bar{V}(t) = \bar{V}^*(t)$. If $\bar{V}(t) = \bar{V}^*(t)$, go to Step 8.

8. Keep $V_{max}(t)$, $V_{min}(t)$, t_{on} and t_{off}.

Notice that, until now, only the values $V_{max}(t)$ and $V_{min}(t)$ have been found. This values guarantee that the average velocity $V^*(t)$ is obtained and therefore the race is finished on time. However, nothing has been said yet about the energetic cost (6.32) of using the control $u(t)$ described by the $V_{max}(t)$, $V_{min}(t)$, t_{on} and t_{off} that have been found. In the following, it is presented an on-line iterative algorithm that will allow to find the $V_{max}(t)$, $V_{min}(t)$, t_{on} and t_{off} such that the smallest cost $C(u(t))_{[t_{on}]}$ is obtained for a gridding in the state-space. Since this iterative algorithm uses Algorithm 6, the values $V_{max}(t)$, $V_{min}(t)$, t_{on} and t_{off} will also guarantee that the race is finished on time.

Algorithm 7

1. Measure $x_1(t)$ at time t and compute $V^*(t)$ using (6.33).
2. Choose a set of n values for $V_{min}(t)$ uniformly spaced from $\bar{V}^*(t)$, as follows $\check{V}_{min} = \{\bar{V}^*(t) - \epsilon, \bar{V}^*(t) - 2\epsilon, \bar{V}^*(t) - 3\epsilon, ..., \bar{V}^*(t) - n\epsilon\}$, with ϵ chosen to have enough elements in $\check{V}_{min}$ to find the smallest cost, e.g. $\epsilon = 1.8$m/s.
3. For each element $V_{min}^{(i)} \in \check{V}_{min}$, $i = 1, ..., n$, perform Algorithm 6 and find $V_{max}^{(i)}$, $V_{min}^{(i)}$, $t_{on}^{(i)}$ and $t_{off}^{(i)}$, $i = 1, ..., n$.
4. For each $t_{on}^{(i)}$, $i = 1, ..., n$, find the cost

$$C(u(t))_{t_{on}^{(i)}} = \frac{C_{on} + V_{oc}(t) I_{batt_{max}} t_{on}^{(i)}}{t_{on}^{(i)}}. \tag{6.38}$$

5. Choose the quantities $V_{max}^{(j)}$, $V_{min}^{(j)}$, $t_{on}^{(j)}$ and $t_{off}^{(j)}$, $j \in \{1, ..., n\}$, such that $C(u(t))_{t_{on}^{(j)}}$ is the smallest cost.
6. Apply the on-off control $u(t)$ described by $V_{max}^{(j)}$, $V_{min}^{(j)}$, $t_{on}^{(j)}$ and $t_{off}^{(j)}$.
7. Return to Step 1.

The overall algorithm that articulates the on-line identification process (Algorithm 5) and the on-line computation of the low consumption driving strategy (Algorithm 6 and Algorithm 7) is shown in Figure 6.2. It was implemented on-board of the *Vir'volt* vehicle during the Shell Eco-Marathon 2014. In the following section, the results obtained during the Shell Eco-Marathon 2014 are presented.

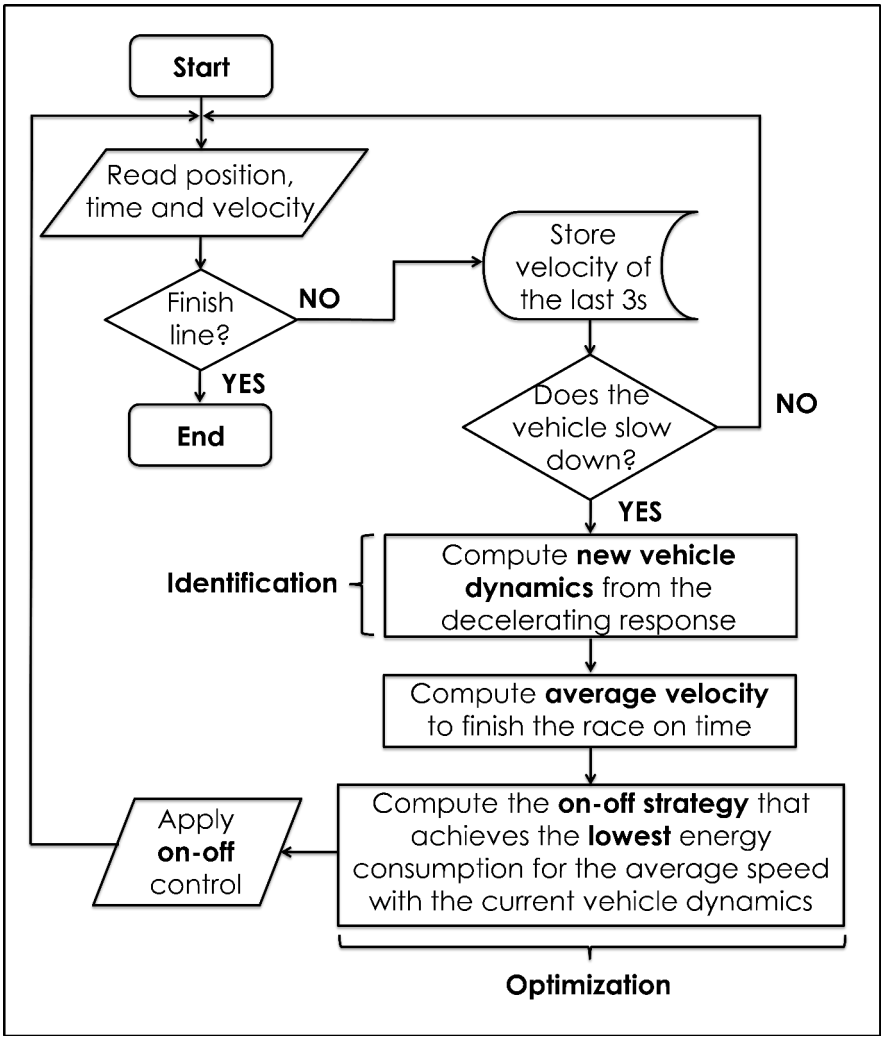


Figure 6.2: Robust adaptive real-time control based on an on-off driving strategy. The total computation time is 22.4ms

6.4 Robust adaptive real-time control based on an on-off driving strategy: application to the Vir'volt vehicle

6.4.1 Off-line identification of the parameter a

The off-line procedure of Algorithm 5, which corresponds to the off-line procedure described in Section 6.2.2.1, has been applied to a long deceleration in order to identify the parameter a involved in the dynamics (6.12). The long deceleration corresponds to a test run realized in April 2014 in the circuit of Saint-Dié-des-Vosges, France. The identification of the parameter a was performed with 30 time samples taken between $t = 690$s and $t = 720$s (see Fig. 6.3), and applying least squares. The following quantities were found

$$\begin{aligned} a &= -0.0010642, \\ b &= -0.0000023, \\ c &= -0.0347565. \end{aligned} \tag{6.39}$$

The identified parameters a, b, and c, were used to reconstruct the velocity $x_2(t)$ using the dynamics (6.2) and under Assumption 5 ($b(t_0) = b(t_1) = ... = b(t_{30})$ and $c(t_0) = c(t_1) = ... = c(t_{30})$). The reconstructed velocity is depicted in Fig. 6.3. The maximal gap between the measured velocity and the reconstructed velocity is 0.13m/s, the maximal relative error (equal to the gap between two velocities divided by the measured velocity) is 2.35%. The average relative error (arithmetical mean of the relatives errors for the 30 samples) is 0.6%.

The good reconstruction performance is explained by two smoothing factors:

1. The first smoothing factor occurs during the solution of the linear system (6.17) for 30 samples in the least square sense. The more samples used, the smoother the result will be.

2. The second smoothing factor lies in the scaling procedure (computation of s^* in Step 2 of the off-line identification in Algorithm 5). The further the first and the last samples are, the best the results will be.

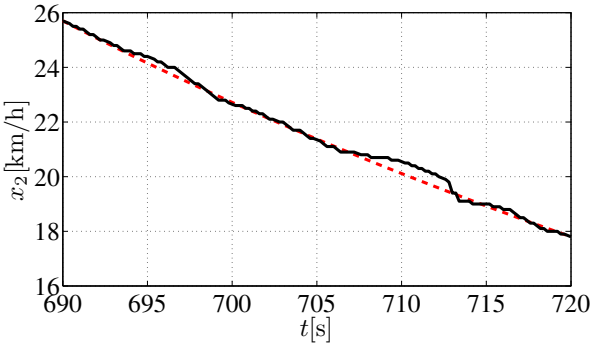


Figure 6.3: Deceleration test performed in Saint-Dié-des-Vosges, France, in April 2014. Black line: measured velocity. Red dashed line: the reconstructed velocity using the parameters identified on-line.

The parameter a in (6.39) is kept. The parameters b and c will be identified on-line. In the following are presented the results obtained when the adaptive algorithm depicted in Fig. 6.2 was applied during the Shell Eco-Marathon performed in May 2014 in Rotterdam.

6.4.2 On-line adaptative real-time control

The control scheme described in Fig. 6.2, was implemented on-board on a microcontroller dsPIC33ep512mu810 of Microchip® (see Section 2.6) during the Shell Eco-Marathon Europe 2014, that took place in Rotterdam, ND, from May 15th to May 18th. The overall on-line algorithm, corresponding to Fig. 6.2, implemented on-board is summarized in the following.

Algorithm 8

On-line steps:

1. Identification of parameters $b(t)$ and $c(t)$ (On-line steps of Algorithm 5): The parameters $b(t)$ and $c(t)$ involved in (6.12) are estimated on-line, at each time t, from the last 3 seconds of deceleration, by making $t_1 - t_0 = t_2 - t_1 = t_3 - t_2 = 1$s and using the parameter a identified off-line and available in (6.39) (see the on-line steps of Algorithm 5).

2. Estimation of the average velocity $V^*(t)$: Algorithm 6 and the identified parameters a and $b(t)$ and $c(t)$ are used to find $\bar{V}^*(t)$, at time t.

3. Estimation of the driving strategy with the smallest energetic consumption: Algorithm 7 is applied to the set of values $\check{V}_{min} = \{\bar{V}^*(t) - 1.8\text{m/s}, \bar{V}^*(t) - 3.6\text{m/s}, \bar{V}^*(t) - 5.4\text{m/s}, ..., \bar{V}^*(t) - 14.4\text{m/s}\}$, with $\epsilon = 1.8\text{m/s}$ and $n = 8$, $C_{on} = 10\text{J}$ and $I_{batt_{max}} = 6.7\text{A}$.

4. The on-off strategy $u(t)$ with the smallest energetic cost $C(u(t))_{t_{on}}$ is applied to the vehicle and the process is repeated from Step 1 until the vehicle reaches to the finish line (see Fig. 6.2).

The on-board implementation is sketched in Fig. 6.4. The standard 32kHz oscillator in the dsPIC, the bike velocity sensor in the *Vir'volt* vehicle and GPS receiver, also in the *Vir'volt* vehicle, provided (respectively) the current time, velocity and position needed by Algorithm 8. The total time that took to the Algorithm 8 to compute the on-off strategy, from Step 1 to Step 4, was 22.4ms. The computational resources (computation time and memory) used by Algorithm 8, embedded in the dsPIC, are reported in Table 6.1. Concerning the ROM, the

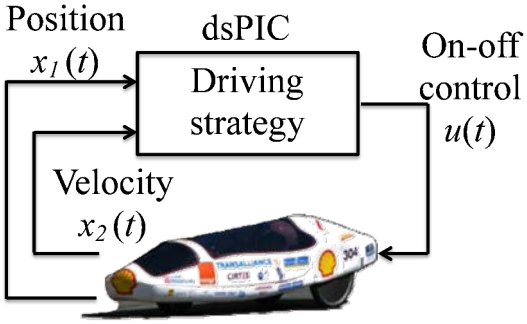


Figure 6.4: On-board implementation of the robust adaptive real-time control during the Shell Eco-Marathon 2014.

Table 6.1: Computational resources of Algorithm 8 embedded in the dsPIC.

ROM	RAM	Loop time
2%	1.53%	22.4ms

results reported in Table 6.1 are smaller than the required ROM memory for the Algorithm 4 (see Table 5.3 in Section 5.8), since Algorithm 8 does not require any look-up table, nor a tracking reference precomputed off-line.

The official results of the EMT prototype measured in the 2014 edition of the Shell Eco-Marathon are presented in Table 6.2, together with the first result obtained in the Shell Eco-Marathon of the year 2013.

Table 6.2: Comparisons of the official results for the *Vir'volt* prototype at Shell Eco-Marathon of years 2013 (manual command) and 2014 (automatic control of Algorithm 8 embedded on-board).

	2013	2014	
	1st run	1st run	2nd run
Run duration			
expected	35m 00s	36m 00s	38m 30s
realized	36m 01s	36m 10s	38m 31s
Consumption	112 742 J	120 150 J	107 946 J

It can be noticed from Table 6.2, that the consumption performances of Algorithm 8 are comparable with performances obtained by human drivers, which were specifically trained to perform a low consumption driving strategy computed off-line. However, during the race in 2014, the driver was able to focus in other aspects of the driving task such as surpassing other vehicles on the track and positioning properly the vehicle before entering in a curve, while the on-board control performed the on-off low consumption driving strategy.

The real-time response of Algorithm 8 (see Fig. 6.2) embedded on-board, is depicted in Fig. 6.5-6.9. It can be observed that the velocity $x_2(t)$ oscillates between $V_{min}(t)$ and $V_{max}(t)$. The average velocity $\bar{V}^*(t)$ is recomputed as $V_{min}(t)$ and $V_{max}(t)$ change in time. In Fig. 6.6, it can be observed that during the acceleration phases, the values of $V_{min}(t)$ and $V_{max}(t)$ remain constant, since they are only recomputed during the deceleration of the vehicle. The continuous actualization of the target average velocity $\bar{V}^*(t)$ makes the strategy of Algorithm 8 extremely robust. Not only the disturbances are taken into account in the computation of the new driving strategy, but also the actual value of the voltage $V_{oc}(t)$ is used to compute the actual cost of the on-off strategy according to the charge remaining in the battery. Thus, robustness is a central outcome provided

by the embedded control.

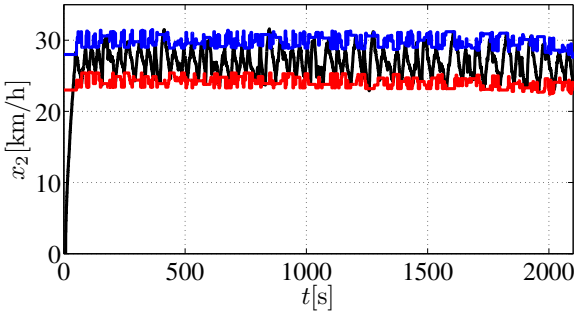


Figure 6.5: Real-time response of the embedded robust adaptive control. Black line: $x_2(t)$. Red line: $V_{min}(t)$. Blue line: $V_{max}(t)$.

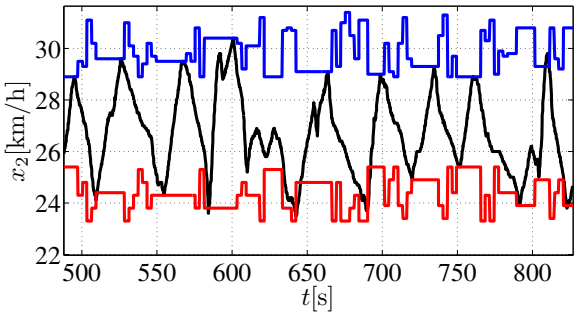


Figure 6.6: Detail of Fig. 6.5 in the interval $t \in [488, 827]$.

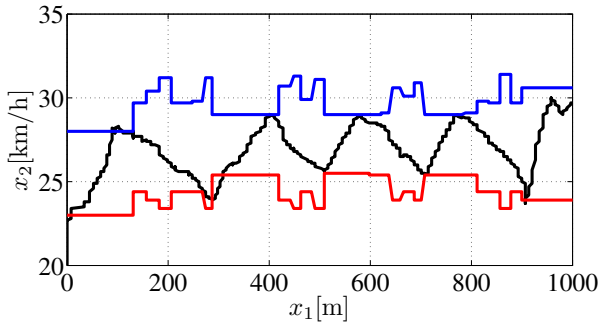


Figure 6.7: Real-time response of the embedded robust adaptive control according to the position. Black line: $x_2(t)$. Red line: $V_{min}(t)$. Blue line: $V_{max}(t)$.

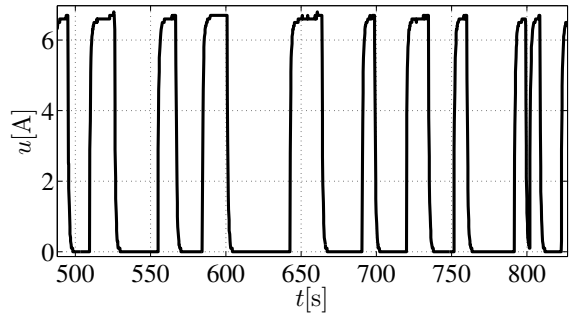


Figure 6.8: Control response $u(t)$ of the embedded robust adaptive control in the interval $t \in [488, 827]$.

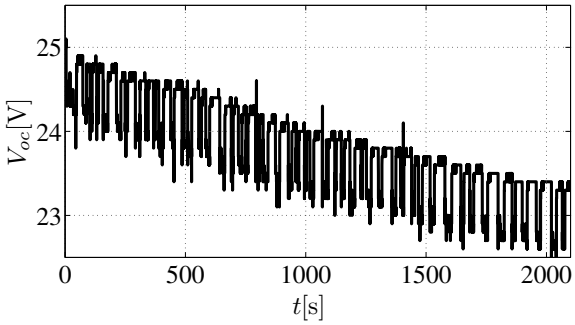


Figure 6.9: Open circuit voltage of the battery V_{oc}. The total variation in the voltage of the battery V_{oc} during all the race, corresponds to a variation of 4.16% of the voltage of the battery fully charged.

6.5 Conclusions

A real-time adaptive driving strategy to achieve a low consumption has been presented. Real life implementation on a competition prototype showed good agreement between the theoretical analysis and the effective results. The consumption performances are comparable with performances obtained by highly trained human drivers. All the computations are done on board, in real-time, with a low cost micro controller, for a limited energy budget.

Among the various advantages of the approach, it can be stressed the dramatic safety improvement for the driver (the driver does not need to concentrate on the velocity) and the robustness of the command that allows effective recovering on optimal timing after traffic perturbations.

Chapter 7

General conclusions and Perspectives

The main objective of this thesis was to propose controlled driving strategies for an electric vehicle prototype that must achieve a minimal energetic consumption. The prototype under consideration is named *Vir'volt* and is involved every year in the Shell Eco-Marathon European race. The main issue, which can be considered as the guiding principle of this work, is to propose controlled driving strategies which can be embedded into the digital devices of the vehicle. As a result, the driving strategies must be compatible with real time constraints, limited memory and computational capacities of the electronic equipment. The computation of the strategies must require itself a low power consumption. Beyond the real-time considerations, the robustness of the control was also a concern.

The main contributions of the work can be sum up as follows:

- A suitable nonlinear model of the electric vehicle has been obtained. The model involved physical equations with parameters estimated from experiments conducted on the vehicle. By suitable, it is meant a model fulfilling the trade-off complexity/precision needed for real-time control purposes.

- The first overall approach has consisted in deriving, first, an optimal driving strategy by solving off-line an optimization problem. The problem amounts to minimizing a cost subject to constraints such as the dynamics of the ve-

hicle, the physical constraints on the vehicle and the race (track profile and length, maximal duration of a run, etc.). As a second step, robust tracking methods of the optimal driving strategy have been proposed. Mainly motivated by their robustness and constraint handling properties, several Model Predictive Control tracking strategies have been detailed.

  - A MPC tracking strategy based on a linearised model around an operating point has been applied to the vehicle in simulation. Time-invariant constraints in the form of polytopic sets, have been considered on the input and the state. The asymptotic stability of the control law has been guaranteed by resorting to an invariant set as an admissible terminal constraint.
  - Then, motivated by the peculiarities induced by the tracking problem, time-varying constraints have been considered, again in a polytopic form. The complexity of the tracking strategy has been preserved, compared to the time-invariant case, by resorting to an homothetic transformation of a nominal invariant set, guaranteeing the asymptotic stability. The resulting MPC tracking strategy has been assessed on the model of the *Vir'volt* vehicle in simulation.
  - To capture the nonlinearities of the dynamics, a LPV model has been further considered. A MPC strategy for LPV systems has been proposed. The contribution that must be pointed out is that the approach is well suited for real-time applications, since it does not involve the on-line solution of any Linear Matrix Inequality (LMI) in the computation of the control law. The LMIs guarantee the stability and the constraints fulfilment. The performances of the approach, in terms of real-time applicability and robustness, have been tested with success on the benchmark for the *Vir'volt* vehicle.

- The principle of the second approach differs from the first one in the sense that the optimal driving strategy is computed on-line so that it can be adapted to a time-varying context. This is precisely the case when there is traffic jam during the race and when phenomena such as wind, rain

and path irregularities are considered. The practical consideration that the efficiency of the power converter may not be optimal on all the operating range has been also taken into account. This motivated an on-off strategy. The resulting on-off adaptive strategy requires an identification performed on-line of the model of the vehicle and of the disturbances. The robust adaptive control had been embedded in a dsPIC device on-board of the *Vir'volt* vehicle, and had been tested with success during the Shell Eco-Marathon 2014. It must be stressed the considerable performances in terms of real-time applicability and robustness.

The publications related to this work are listed below:

- T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. MPC-based tracking for real-time systems subject to time-varying polytopic constraints. In *Optimal Control Applications and Methods (OCAM)*, John Wiley and Sons (2014) (To be published in 2015).

- T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. MPC tracking under time-varying polytopic constraints for real-time applications. In *European Control Conference (ECC)*, 2014. Strasbourg, France, 24 - 27 June 2014. DOI: 10.1109/ECC.2014.6862584.

- T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. MPC for a low-consumption electric vehicle with time-varying constraints. In *IFAC Joint Conference*, 2013. Grenoble, France, 04 - 06 February 2013. DOI: 10.3182/20130204-3-FR-2033.00213.

- T. Manrique, H Malaise, M. Fiacchini, T. Chambrion, and G. Millerioux. Model predictive real-time controller for a low-consumption electric vehicle. In *International Symposium on Environment-Friendly Energies and Applications IEEE (EFEA)*, 2012. Newcastle, UK, 25 - 27 June 2012. DOI: 10.1109/EFEA.2012.6294080. (Best paper award).

Perspectives

The results obtained in this thesis, can be further improved by considering the following alternatives within the tracking task.

- In the tracking task subject to polytopic constraints X_Δ and U_Δ, if the uncertainties in the dynamics of the vehicle, the mismatches between the nonlinear model and the linearised model, and the actual disturbances, all of them being assumed to be bounded within a polytope W, they can be included explicitly in the dynamics through the disturbance w as

$$\Delta x(k+1) = \mathbf{A}\Delta x(k) + \mathbf{B}\Delta u(k) + \mathbf{D}w, \tag{7.1}$$

  where $w \in W$, and $W \in \mathbb{R}^n$ is the set of admissible disturbances. However, since the disturbances are considered to be persistent, then in the best case scenario, the tracking error can be only steered to a neighborhood of the origin. Besides, due to the presence of constraints X_Δ and U_Δ, the disturbances may lead to infeasibility and therefore to instability, if the MPC strategy is solved using only the nominal system (with $w = 0$). Therefore, a robust MPC strategy is required to guarantee feasibility despite the persistent disturbances w. Several approaches tackle this problem, see [24, 54, 31, 46] to mention a few. The main line of those approaches consists in iterating the disturbances forward in time, building a sequence of admissible regions around the nominal prediction (with $w = 0$). Then, the MPC problem is solved with tighter constraints to guarantee that the tracking task fulfils the constraints imposed to the real system (7.1), and thus stability can be guaranteed. This approaches are known as Tube-based MPC. The approaches of [24, 54, 31] may be adapted to the present tracking task and used to include the model mismatches and the nonlinearities of the dynamics. Time-varying polytopic constraints may also be considered using a robust invariant set.

- Regarding the energetic efficiency of the MPC tracking strategy, the MPC problem is continuously recomputed at each sampling time, even if the reference and the problem conditions (disturbances, constraints, etc.) remain

unchanged. This implies that a constant energetic consumption is constantly required by the on-board computation of the control law, even if the control law can be kept constant and no recomputing is required. To tackle this problem, a Self-triggered MPC [15, 29, 28] may be introduced within the closed-loop. This controller allows to compute the MPC strategy only when is necessary, this is, only when the conditions of the task (tracking error, disturbances, etc.) get outside a set considered as the set of acceptable conditions.

- The adaptive real-time control strategy is highly robust, since it identifies on-line the dynamics of the vehicle. Although it is not straightforward, the optimality and the robustness can be formally demonstrated. Moreover, an event-triggered actualization of the strategy, similar to the one mentioned above for the MPC-based tracking, could be implemented to further reduce the computational burden.

- In the actual application, only an electrical source of energy is considered. However, the vehicle can be also equipped with additional energy sources such as solar, hydrogen, fuel, etc. Those energy sources can be used in a combined way to improve the energetic consumption. Once the vehicle has a hybrid energy source, the problem of the optimal driving strategy must be reformulated in order to achieve the minimal consumption. The control strategies must be also reformulated to take into account the hybrid nature of the energy.

Appendix A

Rotterdam's Ahoy circuit

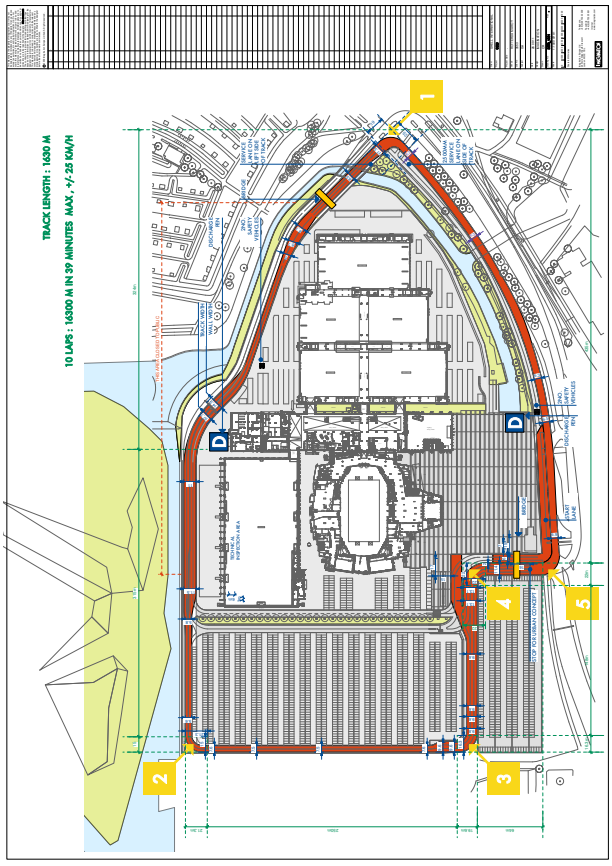


Figure A.1: Rotterdam's Ahoy circuit.

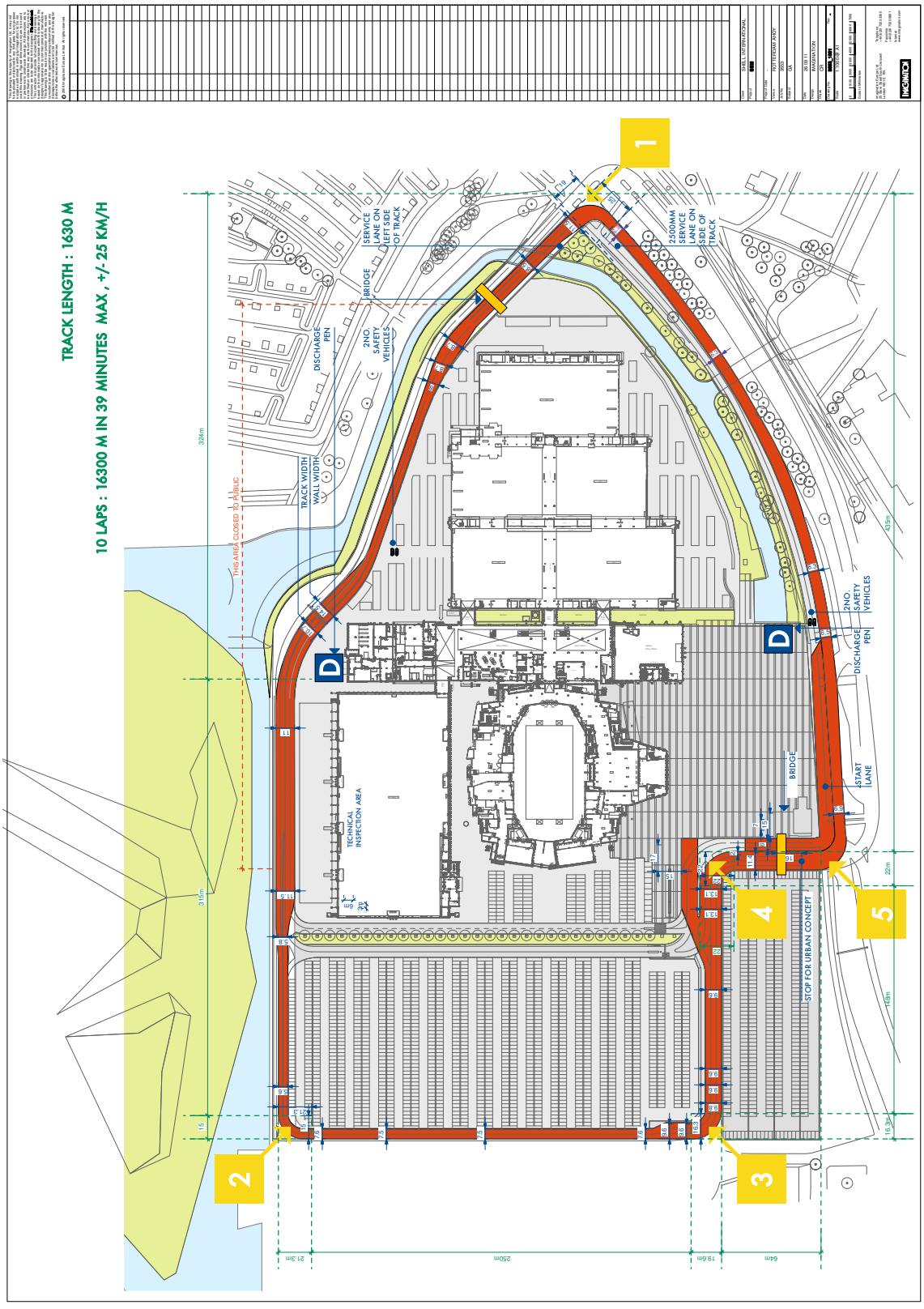


Figure A.2: Detail of Rotterdam's Ahoy circuit.

Appendix B

Proofs of Chapter 5

B.1 Schur complement

The following nonlinear convex inequality

$$Q(x) - S(x)R(x)^{-1}S(x)^T \geq 0, \tag{B.1}$$

where $R(x) > 0$, $Q(x) = Q(x)^T$ and $R(x) = R(x)^T$, is equivalent to the following Linear Matrix Inequality (LMI) (see [19])

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} \geq 0. \tag{B.2}$$

B.2 Proof of equation (5.27)

After expanding (5.18) and (5.19), by means of the Schur complement, it is obtained that

$$\begin{aligned} &\boldsymbol{\Theta}(k) - \gamma(k)^{-1}\left(\mathbf{Q}^{1/2}\boldsymbol{\Theta}(k)\right)^T\left(\mathbf{Q}^{1/2}\boldsymbol{\Theta}(k)\right) - \gamma(k)^{-1}\left(\mathbf{R}^{1/2}\boldsymbol{\Lambda}(k)\right)^T\left(\mathbf{R}^{1/2}\boldsymbol{\Lambda}(k)\right) \\ &\quad - \left(\mathbf{A}_j\boldsymbol{\Theta}(k) + \mathbf{B}\boldsymbol{\Lambda}(k)\right)^T\boldsymbol{\Theta}(k)^{-1}\left(\mathbf{A}_j\boldsymbol{\Theta}(k) + \mathbf{B}\boldsymbol{\Lambda}(k)\right) \geq 0,\ j = 1, 2. \end{aligned} \tag{B.3}$$

Pre and post-multiplying by $(\boldsymbol{\Theta}(k)^{-1})^T$ and $\boldsymbol{\Theta}(k)^{-1}$, respectively, the expression (B.3) becomes

$$\begin{aligned}
&\left(\boldsymbol{\Theta}(k)^{-1}\right)^T \boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1} - \gamma(k)^{-1}\left(\mathbf{Q}^{1/2}\boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1}\right)^T\left(\mathbf{Q}^{1/2}\boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1}\right)\\
&-\gamma(k)^{-1}\left(\mathbf{R}^{1/2}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)^T\left(\mathbf{R}^{1/2}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)\\
&-\left(\mathbf{A}_j\boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1}+\mathbf{B}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)^T\boldsymbol{\Theta}(k)^{-1}\Big(\mathbf{A}_j\boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1}\\
&\qquad\qquad+\mathbf{B}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\Big)\geq 0,\ j=1,2.
\end{aligned} \tag{B.4}$$

Since $\boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1}=\mathbf{I}$, then (B.4), turns into

$$\begin{aligned}
&\boldsymbol{\Theta}(k)^{-1} - \gamma(k)^{-1}\left(\mathbf{Q}^{1/2}\right)^T\left(\mathbf{Q}^{1/2}\right)\\
&-\gamma(k)^{-1}\left(\mathbf{R}^{1/2}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)^T\left(\mathbf{R}^{1/2}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)\\
&-\left(\mathbf{A}_j+\mathbf{B}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)^T\boldsymbol{\Theta}(k)^{-1}\left(\mathbf{A}_j+\mathbf{B}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)\geq 0,\ j=1,2.
\end{aligned} \tag{B.5}$$

Multiplying (B.5) by $(-\gamma(k))$, it is obtained that

$$\begin{aligned}
&-\gamma(k)\boldsymbol{\Theta}(k)^{-1}+\mathbf{Q}+\left(\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)^T\mathbf{R}\left(\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)\\
&+\left(\mathbf{A}_j+\mathbf{B}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)^T\gamma(k)\boldsymbol{\Theta}(k)^{-1}\left(\mathbf{A}_j+\mathbf{B}\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}\right)\leq 0,\ j=1,2.
\end{aligned} \tag{B.6}$$

Notice that if the inequality (B.6) is fulfilled by the vertices of the polytope $\mathcal{C}=Co\{[\mathbf{A}_1\ \mathbf{B}],[\mathbf{A}_2\ \mathbf{B}]\}$, it is also satisfied for all $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}]\in\mathcal{C}$, $i\geq 0$ [42]. Consider now, the gain $F(k)$ that minimizes $\gamma(k)$ in (5.13) and that is defined as

$$F(k)=\boldsymbol{\Lambda}(k)\boldsymbol{\Theta}(k)^{-1}, \tag{B.7}$$

where $\mathbf{\Lambda}(k)$ is solution of (5.16). Reorganising (B.6), and plugging (B.7) into (B.6), gives

$$\Big(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\Big)^T\gamma(k+i)\mathbf{\Theta}(k+i)^{-1}\Big(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\Big) \\ -\gamma(k+i)\mathbf{\Theta}(k+i)^{-1}+F(k+i)^T\mathbf{R}F(k+i)+\mathbf{Q}\leq 0, \tag{B.8}$$

that is satisfied for any $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}]\in\mathcal{C}, i\geq 0$. Replacing $\mathbf{M}(k)=\gamma(k)\mathbf{\Theta}(k)^{-1}$ in (B.8), it is obtained that

$$\Big(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\Big)^T\mathbf{M}(k+i)\Big(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\Big) \\ -\mathbf{M}(k+i)+F(k+i)^T\mathbf{R}F(k+i)+\mathbf{Q}\leq 0. \tag{B.9}$$

Pre and post-multiplying (B.9) by $(\Delta x(k+i))^T$ and $\Delta x(k+i)$, respectively, (B.9) turns into

$$\Big(\Big(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\Big)\Delta x(k+i)\Big)^T\mathbf{M}(k+i)\Big(\Big(\mathbf{A}(\lambda(k+i)) \\ +\mathbf{B}F(k+i)\Big)\Delta x(k+i)\Big)-\Delta x(k+i)^T\mathbf{M}(k+i)\Delta x(k+i) \\ +\Big(F(k+i)\Delta x(k+i)\Big)^T\mathbf{R}\Big(F(k+i)\Delta x(k+i)\Big)+\Delta x(k+i)^T\mathbf{Q}\Delta x(k+i)\leq 0. \tag{B.10}$$

Notice that $\Delta x(k+i+1)=\left(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\right)\Delta x(k+i)$, in virtue of the the control law $\Delta u(k+i)=F(k+i)\Delta x(k+i)$. Therefore, (B.10) becomes

$$\Delta x(k+i+1)^T\mathbf{M}(k+i)\Delta x(k+i+1)-\Delta x(k+i)^T\mathbf{M}(k+i)\Delta x(k+i) \\ +\Delta u(k+i)^T\mathbf{R}\Delta u(k+i)+\Delta x(k+i)^T\mathbf{Q}\Delta x(k+i)\leq 0. \tag{B.11}$$

Using (5.23), (B.11) is rewritten as

$$V(\Delta x(k+i+1))-V(\Delta x(k+i))+\Delta u(k+i)^T\mathbf{R}\Delta u(k+i) \\ +\Delta x(k+i)^T\mathbf{Q}\Delta x(k+i)\leq 0, \tag{B.12}$$

which is equivalent to

$$V(\Delta x(k+i+1)) - V(\Delta x(k+i)) \leq \\ - \Big(\Delta x(k+i)^T \mathbf{Q} \Delta x(k+i) + \Delta u(k+i)^T \mathbf{R} \Delta u(k+i)\Big). \tag{B.13}$$

The expression (B.13) is satisfied for all $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$, $i \geq 0$.

B.3 Proof of equation (5.28)

Summing (B.13) from zero to infinity, yields

$$\sum_{i=0}^{\infty} \Big(V(\Delta x(k+i+1)) - V(\Delta x(k+i))\Big) \\ \leq -\sum_{i=0}^{\infty} \Big(\Delta x(k+i)^T \mathbf{Q} \Delta x(k+i) + \Delta u(k+i)^T \mathbf{R} \Delta u(k+i)\Big). \tag{B.14}$$

The left side of (B.14) is developed as follows

$$\begin{aligned}
&\sum_{i=0}^{\infty}\Big(V(\Delta x(k+i+1)) - V(\Delta x(k+i))\Big) \\
&\quad= \sum_{i=0}^{\infty} V(\Delta x(k+i+1)) - \sum_{i=0}^{\infty} V(\Delta x(k+i)), \\
&\quad= \sum_{i=0}^{\infty}\Big(\Delta x(k+i+1)^T\mathbf{M}(k+i+1)\Delta x(k+i+1)\Big) \\
&\qquad\qquad - \sum_{i=0}^{\infty}\Big(\Delta x(k+i)^T\mathbf{M}(k+i)\Delta x(k+i)\Big), \\
&\quad= \sum_{i=1}^{\infty}\Big(\Delta x(k+i)^T\mathbf{M}(k+i)\Delta x(k+i)\Big) \\
&\qquad\qquad - \sum_{i=0}^{\infty}\Big(\Delta x(k+i)^T\mathbf{M}(k+i)\Delta x(k+i)\Big), \\
&\quad= \sum_{i=1}^{\infty}\Big(\Delta x(k+i)^T\mathbf{M}(k+i)\Delta x(k+i)\Big) - \Big(\big(\Delta x(k)^T\mathbf{M}(k)\Delta x(k)\big) \\
&\qquad\qquad + \sum_{i=1}^{\infty}\Delta x(k+i)^T\mathbf{M}(k+i)\Delta x(k+i)\Big), \\
&\quad= -\Delta x(k)^T\mathbf{M}(k)\Delta x(k).
\end{aligned} \tag{B.15}$$

Plugging (B.15) into (B.14), (B.14) becomes

$$\begin{aligned}
-\Delta x(k)^T\mathbf{M}(k)\Delta x(k) \leq -\sum_{i=0}^{\infty}\Big(&\Delta x(k+i)^T\mathbf{Q}\Delta x(k+i) \\
&+ \Delta u(k+i)^T\mathbf{R}\Delta u(k+i)\Big).
\end{aligned} \tag{B.16}$$

Finally, using (5.12), (B.16) turns into

$$\begin{aligned}
-\Delta x(k)^T\mathbf{M}(k)\Delta x(k) &\leq -J(k), \\
-V(\Delta x(k)) &\leq -J(k), \\
J(k) &\leq V(\Delta x(k)).
\end{aligned} \tag{B.17}$$

B.4 Proof of equation (5.31)

Applying the Schur complement to (5.20), the following is obtained

$$\boldsymbol{\Theta}(k) - \boldsymbol{\Lambda}(k)^T \frac{1}{\Delta u_{max}^2} \boldsymbol{\Lambda}(k) \geq 0. \tag{B.18}$$

If $\boldsymbol{\Lambda}(k) = F(k)\boldsymbol{\Theta}(k)$ (from (5.24)) is plugged into (B.18), it turns into

$$\begin{aligned} \boldsymbol{\Theta}(k) - \frac{1}{\Delta u_{max}^2} (F(k)\boldsymbol{\Theta}(k))^T F(k)\boldsymbol{\Theta}(k) \geq 0, \\ \boldsymbol{\Theta}(k) - \frac{1}{\Delta u_{max}^2} F(k)^T \boldsymbol{\Theta}(k)^T F(k)\boldsymbol{\Theta}(k) \geq 0. \end{aligned} \tag{B.19}$$

Pre and post-multiplying (B.19) by $(\boldsymbol{\Theta}(k)^{-1})^T$ and $\boldsymbol{\Theta}(k)^{-1}$, respectively, gives

$$\begin{aligned} (\boldsymbol{\Theta}(k)^{-1})^T \boldsymbol{\Theta}(k) \boldsymbol{\Theta}(k)^{-1} - \frac{1}{\Delta u_{max}^2} (\boldsymbol{\Theta}(k)^{-1})^T (F(k)\boldsymbol{\Theta}(k))^T F(k)\boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1} \geq 0, \\ (\boldsymbol{\Theta}(k)^{-1})^T - \frac{1}{\Delta u_{max}^2} (F(k)\boldsymbol{\Theta}(k)\boldsymbol{\Theta}(k)^{-1})^T F(k) \geq 0, \\ (\boldsymbol{\Theta}(k)^{-1})^T - \frac{1}{\Delta u_{max}^2} F(k)^T F(k) \geq 0. \end{aligned} \tag{B.20}$$

After pre and post-multiplying (B.20) by $(\Delta x(k))^T$ and $\Delta x(k)$, respectively, the following is obtained

$$\begin{aligned} (\Delta x(k))^T (\boldsymbol{\Theta}(k)^{-1})^T \Delta x(k) - \frac{1}{\Delta u_{max}^2} (\Delta x(k))^T F(k)^T F(k) \Delta x(k) \geq 0, \\ (\boldsymbol{\Theta}(k)^{-1} \Delta x(k))^T \Delta x(k) - \frac{1}{\Delta u_{max}^2} (F(k)\Delta x(k))^T F(k) \Delta x(k) \geq 0, \\ (\Delta x(k)^T \boldsymbol{\Theta}(k)^{-1} \Delta x(k))^T - \frac{1}{\Delta u_{max}^2} (F(k)\Delta x(k))^T F(k) \Delta x(k) \geq 0. \end{aligned} \tag{B.21}$$

Reorganising (B.21), letting $\Delta u(k) = F(k)\Delta x(k)$, $\boldsymbol{\Theta}(k)^{-1} = \frac{1}{\gamma(k)}\mathbf{M}(k)$, and $V(\Delta x(k)) = \Delta x^T(k)\mathbf{M}(k)\Delta x(k)$, it is obtained that

$$\begin{aligned}
\frac{1}{\Delta u_{max}^2}(F(k)\Delta x(k))^T F(k)\Delta x(k) &\leq (\Delta x(k)^T \boldsymbol{\Theta}(k)^{-1}\Delta x(k))^T, \\
\frac{1}{\Delta u_{max}^2}(\Delta u(k))^T \Delta u(k) &\leq \left(\Delta x(k)^T \frac{1}{\gamma(k)}\mathbf{M}(k)\Delta x(k)\right)^T, \\
\frac{1}{\Delta u_{max}^2}(\Delta u(k))^T \Delta u(k) &\leq \frac{1}{\gamma(k)}(\Delta x(k)^T \mathbf{M}(k)\Delta x(k))^T, \\
\frac{1}{\Delta u_{max}^2}(\Delta u(k))^T \Delta u(k) &\leq \frac{1}{\gamma(k)}V(\Delta x(k))^T, \\
\frac{1}{\Delta u_{max}^2}(\Delta u(k))^T \Delta u(k) &\leq \frac{1}{\gamma(k)}V(\Delta x(k)).
\end{aligned} \tag{B.22}$$

Notice that in virtue of (5.25), the following stands

$$\begin{aligned}
V(\Delta x(k)) &\leq \gamma(k), \\
\frac{1}{\gamma(k)}V(\Delta x(k)) &\leq 1.
\end{aligned} \tag{B.23}$$

Therefore, using (B.23), (B.22) becomes

$$\begin{aligned}
\frac{1}{\Delta u_{max}^2}(\Delta u(k))^T \Delta u(k) \leq \frac{1}{\gamma(k)}V(\Delta x(k)) \leq 1, \\
(\Delta u(k))^T \Delta u(k) \leq \Delta u_{max}^2.
\end{aligned} \tag{B.24}$$

B.5 Proof of equation (5.32)

Consider the matrix inequalities (5.21) and (5.22) at time $k+i$, $i \geq 0$, i.e.

$$\begin{bmatrix} \Delta y_{max}^2 \boldsymbol{I} & \star \\ (\mathbf{A}_j\boldsymbol{\Theta}(k+i) + \mathbf{B}\boldsymbol{\Lambda}(k+i))^T \mathbf{C}^T & \boldsymbol{\Theta}(k+i) \end{bmatrix} \geq 0, \tag{B.25}$$

with $j = 1, 2$. Applying the Schur complement to the matrix inequality (B.25), the following is obtained

$$\boldsymbol{\Theta}(k+i) - (\mathbf{A}_j\boldsymbol{\Theta}(k+i) + \mathbf{B}\boldsymbol{\Lambda}(k+i))^T \mathbf{C}^T \frac{1}{\Delta y_{max}^2}\mathbf{C}\,(\mathbf{A}_j\boldsymbol{\Theta}(k+i) + \mathbf{B}\boldsymbol{\Lambda}(k+i)) \geq 0. \tag{B.26}$$

Recall that since the inequality (B.26) is fulfilled by the vertices of the polytope $\mathcal{C} = Co\{[\mathbf{A}_1\ \mathbf{B}], [\mathbf{A}_2\ \mathbf{B}]\}$, it is also satisfied for all $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$, $i \geq 0$ [42]. Thus,

$$\begin{aligned}\boldsymbol{\Theta}(k+i) - \left(\mathbf{A}(\lambda(k+i))\boldsymbol{\Theta}(k+i) + \mathbf{B}\boldsymbol{\Lambda}(k+i)\right)^T \mathbf{C}^T \frac{1}{\Delta y_{max}^2} \\ \mathbf{C}\left(\mathbf{A}(\lambda(k+i))\boldsymbol{\Theta}(k+i) + \mathbf{B}\boldsymbol{\Lambda}(k+i)\right) \geq 0,\end{aligned} \tag{B.27}$$

where $[\mathbf{A}(\lambda(k+i))\ \mathbf{B}] \in \mathcal{C}$. Plugging $\boldsymbol{\Lambda}(k+i) = F(k+i)\boldsymbol{\Theta}(k+i)$ into (B.27), gives

$$\begin{aligned}\boldsymbol{\Theta}(k+i) - \left(\mathbf{A}(\lambda(k+i))\boldsymbol{\Theta}(k+i) + \mathbf{B}F(k+i)\boldsymbol{\Theta}(k+i)\right)^T \mathbf{C}^T \frac{1}{\Delta y_{max}^2} \\ \mathbf{C}\left(\mathbf{A}(\lambda(k+i))\boldsymbol{\Theta}(k) + \mathbf{B}F(k)\boldsymbol{\Theta}(k+i)\right) \geq 0,\end{aligned} \tag{B.28}$$

which is equivalent to

$$\begin{aligned}\boldsymbol{\Theta}(k+i) - \frac{1}{\Delta y_{max}^2}\left(\mathbf{A}(\lambda(k+i))\boldsymbol{\Theta}(k+i) + \mathbf{B}F(k+i)\boldsymbol{\Theta}(k+i)\right)^T \mathbf{C}^T \\ \mathbf{C}\left(\mathbf{A}(\lambda(k))\boldsymbol{\Theta}(k+i) + \mathbf{B}F(k+i)\boldsymbol{\Theta}(k)\right) \geq 0.\end{aligned} \tag{B.29}$$

Pre and post-multiplying (B.29) by $(\boldsymbol{\Theta}(k+i)^{-1})^T$ and $\boldsymbol{\Theta}(k+i)^{-1}$, respectively, gives

$$\begin{aligned}&(\boldsymbol{\Theta}(k+i)^{-1})^T\boldsymbol{\Theta}(k+i)\boldsymbol{\Theta}(k+i)^{-1} \\ &- \frac{1}{\Delta y_{max}^2}(\boldsymbol{\Theta}(k+i)^{-1})^T\left(\mathbf{A}(\lambda(k))\boldsymbol{\Theta}(k+i) + \mathbf{B}F(k+i)\boldsymbol{\Theta}(k+i)\right)^T \mathbf{C}^T \\ &\qquad \mathbf{C}\left(\mathbf{A}(\lambda(k+i))\boldsymbol{\Theta}(k+i) + \mathbf{B}F(k+i)\boldsymbol{\Theta}(k+i)\right)\boldsymbol{\Theta}(k+i)^{-1} \geq 0,\end{aligned} \tag{B.30}$$

which can be rewritten as

$$\begin{aligned}(\boldsymbol{\Theta}(k+i)^{-1})^T - \frac{1}{\Delta y_{max}^2}\left(\mathbf{A}(\lambda(k+i)) + \mathbf{B}F(k+i)\right)^T \mathbf{C}^T \\ \mathbf{C}\left(\mathbf{A}(\lambda(k+i)) + \mathbf{B}F(k+i)\right) \geq 0.\end{aligned} \tag{B.31}$$

Pre and post-multiplying (B.31) by $(\Delta x(k+i))^T$ and $\Delta x(k+i)$, respectively, gives

$$\begin{aligned}(\Delta x(k+i))^T(\boldsymbol{\Theta}(k+i)^{-1})^T\Delta x(k+i)\\ -\frac{1}{\Delta y_{max}^2}(\Delta x(k+i))^T\left(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\right)^T\mathbf{C}^T\\ \mathbf{C}\left(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k)\right)\Delta x(k+i)\geq 0.\end{aligned} \tag{B.32}$$

Expression (B.32) is equivalent to

$$\begin{aligned}(\Delta x(k+i))^T(\boldsymbol{\Theta}(k+i)^{-1})^T\Delta x(k+i)\\ -\frac{1}{\Delta y_{max}^2}\Big(\left(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\right)\Delta x(k+i)\Big)^T\mathbf{C}^T\\ \mathbf{C}\Big(\left(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k+i)\right)\Delta x(k+i)\Big)\geq 0.\end{aligned} \tag{B.33}$$

Recall that $\left(\mathbf{A}(\lambda(k+i))+\mathbf{B}F(k)\right)\Delta x(k+i)=\Delta x(k+i+1)$. Therefore (B.33), turns into

$$(\Delta x(k+i))^T(\boldsymbol{\Theta}(k+i)^{-1})^T\Delta x(k+i)-\frac{1}{\Delta y_{max}^2}\left(\Delta x(k+i+1)\right)^T\mathbf{C}^T\mathbf{C}\Delta x(k+i+1)\geq 0, \tag{B.34}$$

which is equivalent to

$$(\Delta x(k+i))^T(\boldsymbol{\Theta}(k+i)^{-1})^T\Delta x(k+i)-\frac{1}{\Delta y_{max}^2}\left(\mathbf{C}\Delta x(k+i+1)\right)^T\mathbf{C}\Delta x(k+i+1)\geq 0. \tag{B.35}$$

Since $\Delta y(k+i+1)=\mathbf{C}\Delta x(k+i+1)$, equation (B.35) becomes

$$(\Delta x(k+i))^T(\boldsymbol{\Theta}(k)^{-1})^T\Delta x(k+i)-\frac{1}{\Delta y_{max}^2}\left(\Delta y(k+i+1)\right)^T\Delta y(k+i+1)\geq 0. \tag{B.36}$$

Reorganizing (B.36), gives

$$\frac{1}{\Delta y_{max}^2}\left(\Delta y(k+i+1)\right)^T\Delta y(k+i+1)\leq(\Delta x(k+i))^T(\boldsymbol{\Theta}(k+i)^{-1})^T\Delta x(k+i), \tag{B.37}$$

which is equivalent to

$$\frac{1}{\Delta y_{max}^2}\left(\Delta y(k+i+1)\right)^T \Delta y(k+i+1) \leq \left(\Delta x(k+i)^T \boldsymbol{\Theta}(k+i)^{-1} \Delta x(k+i)\right)^T. \quad \text{(B.38)}$$

Since $\boldsymbol{\Theta}(k+i)^{-1} = \frac{1}{\gamma(k+i)}\mathbf{M}(k+i)$ and $V(\Delta x(k+i)) = \Delta x^T(k+i)\mathbf{M}(k+i)\Delta x(k+i)$, at time $k+i$, (B.38) turns into

$$\begin{aligned}\frac{1}{\Delta y_{max}^2}\left(\Delta y(k+i+1)\right)^T \Delta y(k+i+1) &\leq \left(\Delta x(k+i)^T \frac{1}{\gamma(k+i)}\mathbf{M}(k+i)\Delta x(k+i)\right)^T \\ &= \frac{1}{\gamma(k+i)}\left(\Delta x(k+i)^T \mathbf{M}(k+i)\Delta x(k+i)\right)^T, \\ &= \frac{1}{\gamma(k+i)}\left(V(\Delta x(k+i))\right)^T, \\ &= \frac{1}{\gamma(k+i)} V(\Delta x(k+i)).\end{aligned} \quad \text{(B.39)}$$

Plugging (B.23) into (B.39), gives

$$\begin{aligned}\frac{1}{\Delta y_{max}^2}\left(\Delta y(k+i+1)\right)^T \Delta y(k+i+1) &\leq 1, \\ \left(\Delta y(k+i+1)\right)^T \Delta y(k+i+1) &\leq \Delta y_{max}^2,\end{aligned} \quad \text{(B.40)}$$

with $i \geq 0$. Thus,

$$\left(\Delta y(k+i)\right)^T \Delta y(k+i) \leq \Delta y_{max}^2, \ \forall i \geq 1. \quad \text{(B.41)}$$

Appendix C

Proofs of Chapter 6

C.1 Proof of equation (6.13)

Consider the deceleration dynamics (6.12), i.e.

$$\frac{dx_2(t)}{dt} = ax_2(t)^2 + b(t)x_2(t) + c(t). \tag{C.1}$$

Notice that (C.1) can be rewritten as

$$\begin{aligned} \frac{\frac{dx_2(t)}{dt}}{ax_2(t)^2 + b(t)x_2(t) + c(t)} &= 1, \\ \frac{dx_2(t)}{ax_2(t)^2 + b(t)x_2(t) + c(t)} &= dt. \end{aligned} \tag{C.2}$$

If (C.2) is integrated in an interval of time $\tau = [t_0, t]$, in which $b(\tau)\Big|_{\tau=t_0}^{\tau=t} = b$ and $c(\tau)\Big|_{\tau=t_0}^{\tau=t} = c$, then

$$\int_{y=x_2(t_0)}^{y=x_2(t)} \frac{\mathrm{d}y}{ay^2 + by + c} = \int_{\tau=t_0}^{\tau=t} \mathrm{d}\tau, \tag{C.3}$$

$$\int_{x_2(t_0)}^{x_2(t)} \frac{\mathrm{d}y}{ay^2 + by + c} = t - t_0. \tag{C.4}$$

Factorizing $ay^2 + by + c$, it is obtained that

$$ay^2 + by + c = a\left(y - \frac{-b + \sqrt{b^2 - 4ac}}{2a}\right)\left(y - \frac{-b - \sqrt{b^2 - 4ac}}{2a}\right). \tag{C.5}$$

Plugging (C.5) into (C.4), the left side of (C.4) becomes

$$\int_{x_2(t_0)}^{x_2(t)} \frac{\mathrm{d}y}{ay^2+by+c} = \frac{1}{\sqrt{b^2-4\,a\,c}}\left(\log\left(\frac{2\,a\,x_2(t)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t)+\sqrt{b^2-4\,a\,c}+b}\right)\right.$$
$$\left. - \log\left(\frac{2\,a\,x_2(t_0)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_0)+\sqrt{b^2-4\,a\,c}+b}\right)\right). \tag{C.6}$$

Consider the time instants t_0, t_1 and t_2, in which the measurements of the velocity $x_2(t_0)$, $x_2(t_1)$ and $x_2(t_2)$ are performed. It is assumed that $b(t_0) = b(t_1) = b(t_2) = b$ and $c(t_0) = c(t_1) = c(t_2) = c$. Additionally t_0, t_1 and t_2 are uniformly spaced, i.e. $t_2 - t_1 = t_1 - t_0$. In virtue of (C.6), it is obtained that

$$\log\left(\frac{2\,a\,x_2(t_1)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_1)+\sqrt{b^2-4\,a\,c}+b}\right) - \log\left(\frac{2\,a\,x_2(t_0)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_0)+\sqrt{b^2-4\,a\,c}+b}\right)$$
$$= \log\left(\frac{2\,a\,x_2(t_2)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_2)+\sqrt{b^2-4\,a\,c}+b}\right) - \log\left(\frac{2\,a\,x_2(t_1)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_1)+\sqrt{b^2-4\,a\,c}+b}\right). \tag{C.7}$$

Recalling that for any $a, b, c, d > 0$,

$$\log(a) - \log(b) = \log(c) - \log(d) \Leftrightarrow \log(a/b) = \log(c/d) \Leftrightarrow ad = bc, \tag{C.8}$$

then (C.7) is equivalent to

$$\frac{2\,a\,x_2(t_1)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_1)+\sqrt{b^2-4\,a\,c}+b} \times \frac{2\,a\,x_2(t_1)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_1)+\sqrt{b^2-4\,a\,c}+b}$$
$$= \frac{2\,a\,x_2(t_2)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_2)+\sqrt{b^2-4\,a\,c}+b} \times \frac{2\,a\,x_2(t_0)-\sqrt{b^2-4\,a\,c}+b}{2\,a\,x_2(t_0)+\sqrt{b^2-4\,a\,c}+b}. \tag{C.9}$$

Introducing the auxiliary variables $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\gamma}$, defined as follows

$$\begin{aligned} \hat{\alpha} &= 2a, \\ \hat{\beta} &= \sqrt{b^2-4\,a\,c}+b, \\ \hat{\gamma} &= -\sqrt{b^2-4\,a\,c}+b, \end{aligned} \tag{C.10}$$

the expression (C.9) can be rewritten as

$$\left(\frac{\hat{\alpha}x_2(t_1)+\hat{\gamma}}{\hat{\alpha}x_2(t_1)+\hat{\beta}}\right)^2 = \frac{\hat{\alpha}x_2(t_2)+\hat{\gamma}}{\hat{\alpha}x_2(t_2)+\hat{\beta}} \times \frac{\hat{\alpha}x_2(t_0)+\hat{\gamma}}{\hat{\alpha}x_2(t_0)+\hat{\beta}}. \tag{C.11}$$

The expression (C.11) is equivalent to

$$\begin{aligned}\left(\hat{\alpha}x_2(t_1)+\hat{\gamma}\right)^2 \left(\hat{\alpha}x_2(t_2)+\hat{\beta}\right) \left(\hat{\alpha}x_2(t_0)+\hat{\beta}\right) \\ = \left(\hat{\alpha}x_2(t_1)+\hat{\beta}\right)^2 \left(\hat{\alpha}x_2(t_0)+\hat{\gamma}\right) \left(\hat{\alpha}x_2(t_2)+\hat{\gamma}\right).\end{aligned} \tag{C.12}$$

Simplifying (C.12), it becomes

$$\begin{aligned}&\left[x_2(t_1)^2 \left(x_2(t_0)+x_2(t_2)\right) - 2x_2(t_0)x_2(t_1)x_2(t_2)\right] \hat{\alpha}^2 \\ &\quad + \left(x_2(t_1)^2 + x_2(t_0)x_2(t_2)\right) \hat{\alpha} \left(\hat{\beta}+\hat{\gamma}\right) + \left[2x_2(t_1) - \left(x_2(t_0)+x_2(t_2)\right)\right] \hat{\beta}\hat{\gamma} = 0.\end{aligned} \tag{C.13}$$

Notice that from (C.10), $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\gamma}$ verify

$$\begin{aligned}\hat{\alpha}^2 &= 4a^2, \\ \hat{\alpha}\left(\hat{\beta}+\hat{\gamma}\right) &= 4ab, \\ \hat{\beta}\hat{\gamma} &= 4ac.\end{aligned} \tag{C.14}$$

Therefore, plugging (C.14) into (C.13), gives

$$\begin{aligned}&\left[x_2(t_1)^2 \left(x_2(t_0)+x_2(t_2)\right) - 2x_2(t_0)x_2(t_1)x_2(t_2)\right] 4a^2 \\ &\quad + \left(x_2(t_1)^2 + x_2(t_0)x_2(t_2)\right) 4ab + \left[2x_2(t_1) - \left(x_2(t_0)+x_2(t_2)\right)\right] 4ac = 0.\end{aligned} \tag{C.15}$$

Factorizing $4a$, (C.15) becomes

$$\begin{aligned}&\left[x_2(t_1)^2 \left(x_2(t_0)+x_2(t_2)\right) - 2x_2(t_0)x_2(t_1)x_2(t_2)\right] a \\ &\quad + \left(x_2(t_1)^2 + x_2(t_0)x_2(t_2)\right) b + \left[2x_2(t_1) - \left(x_2(t_0)+x_2(t_2)\right)\right] c = 0.\end{aligned} \tag{C.16}$$

References

[1] EcoCar 2. http://www.ecocar2.org/about-ecocar2. [Online; accessed 05-August-2014].

[2] EcoMotion Team. http://www.ecomotionteam.org/blog/. [Online; accessed 05-August-2014].

[3] Shell Eco-Marathon. http://www.shell.com/global/environment-society/ecomarathon.html. [Online; accessed 05-August-2014].

[4] Shell Eco-marathon 2015 Official Rules. http://s02.static-shell.com/content/dam/shell-new/local/corporate/ecomarathon/downloads/pdf/global/sem-2015-global-rules-chapter1-010714.pdf. [Online; accessed 05-August-2014].

[5] Shell Eco-marathon Flickr. https://www.flickr.com/photos/shell_eco-marathon/. [Online; accessed 05-August-2014].

[6] WAVE (World Advanced Vehicle Expedition) Throphy. http://www.wavetrophy.com/en/. [Online; accessed 05-August-2014].

[7] Zero Emissions Race. http://www.zero-race.com/en/. [Online; accessed 05-August-2014].

[8] G. Acampora, C. Landi, M. Luiso, and N. Pasquino. Optimization of energy consumption in a railway traction system. In *Power Electronics, Electrical Drives, Automation and Motion, 2006. SPEEDAM 2006. International Symposium on*, pages 1121–1126, May 2006.

[9] A. A. Agrachev and Y. L. Sachkov. *Control theory from the geometric viewpoint*, **87** of *Encyclopaedia of Mathematical Sciences*. Springer-Verlag, Berlin, 2004. Control Theory and Optimization, II.

[10] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In *Nonlinear Model Predictive Control*, **384** of *Lecture Notes in Control and Information Sciences*, pages 345–369. Springer Berlin Heidelberg, 2009.

[11] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, **245** of *Lecture Notes in Control and Information Sciences*, pages 207–226. Springer London, 1999.

[12] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, **38**[1]:3–20, January 2002.

[13] S.C. Bengea and R.A Decarlo. Optimal and suboptimal control of switching systems. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, **5**, pages 5295–5300 Vol.5, Dec 2003.

[14] Sorin C. Bengea and Raymond A. DeCarlo. Optimal control of switching systems. *Automatica J. IFAC*, **41**[1]:11–27, 2005.

[15] J.D.J. Barradas Berglind, T.M.P. Gommas, and W.P.M.H. Heemels. Self-triggered MPC for constrained linear systems and quadratic costs. In *4th IFAC Nonlinear Model Predictive Control Conference*, Noordwijkerhout, NL, 2012.

[16] F. Blanchini. Set invariance in control. *Automatica*, **35**:1747–1767, 1999.

[17] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Birkhäuser, 2008.

[18] C. Bordons, M. A. Ridao, A. Pérez, A. Arce, and D. Marcos. Model Predictive Control for power management in hybrid fuel cell vehicles. In *Vehicle Power and Propulsion IEEE Conference*, 2010.

[19] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, **15** of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, jun 1994.

[20] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory. *Automatic Control, IEEE Transactions on*, **43**[1]:31–45, Jan 1998.

[21] F. Bruzelius. *Linear Parameter-Varying Systems: an approach to gain scheduling*. PhD thesis, Chalmers University of Technology, 2004.

[22] P. Bumroongsri and S. Kheawhom. MPC for LPV systems based on parameter-dependent lyapunov function with perturbation on control input strategy. *Engineering Journal*, **16**[2], 2012.

[23] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer-Verlag, 2004.

[24] L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: Predictive control with restricted constraints. *Automatica*, **37**:1019–1028, 2001.

[25] X. C. Ding, Y. Wardi, and M. Egerstedt. On-line optimization of switched-mode dynamical systems. *Automatic Control, IEEE Transactions on*, **54**[9]:2266–2271, Sept 2009.

[26] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-time optimization for switched-mode dynamical systems. *Automatic Control, IEEE Transactions on*, **51**[1]:110–115, Jan 2006.

[27] M. Ehsani, Y. Gao, S. E. Gay, and A. Emadi. *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory and Design*. CRC Press, 2004.

[28] A Eqtami, D.V. Dimarogonas, and K.J. Kyriakopoulos. Event-triggered control for discrete-time systems. In *American Control Conference (ACC), 2010*, pages 4719–4724, June 2010.

[29] A Eqtami, D.V. Dimarogonas, and K.J. Kyriakopoulos. Novel event-triggered strategies for model predictive controllers. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 3392–3397, Dec 2011.

[30] E. G. Gilbert and K. T. Tan. Linear systems with state and control constrains: The theory and application of maximal output admissible sets. *Transactions on automatic control*, **36**:1008–1020, 1991.

[31] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodriguez. Online robust tube-based MPC for time-varying systems: a practical approach. *International Journal of Control*, **84**[6]:1157–1170, June 2011.

[32] L. Guzzella and A. Sciarretta. *Vehicle propulsion systems : introduction to modeling and optimization*. Springer, Berlin ; New York, 2005.

[33] F. Haugen. *Discrete-time signals and systems*. TechTeach, 2005.

[34] J. Hauth. *Grey-Box Modelling for Nonlinear Systems*. PhD thesis, TU Kaiserslautern, 2008.

[35] S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, **4**, pages 3972–3977 vol.4, 1999.

[36] A. V. Rao; D. Benson; C. Darby; G. T. Huntington. User's Manual for GPOPS Version 4.x: A MATLAB® Software for Solving Multi-Phase Optimal Control Problems Using hp-Adaptative Pseudospectral Methods, 2011.

[37] Shengxiang Jiang and P.G. Voulgaris. Performance optimization of switched systems: A model matching approach. *Automatic Control, IEEE Transactions on*, **54**[9]:2058–2071, Sept 2009.

[38] E.R. Johnson and T.D. Murphey. Second-order switching time optimization for nonlinear time-varying dynamic systems. *Automatic Control, IEEE Transactions on*, **56**[8]:1953–1957, Aug 2011.

[39] R.E. Kalman. Contributions to the theory of optimal control. *Boletin Sociedad Matematica Mexicana*, [5]:102–119.

[40] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete time linear systems. *Mathematical Problems in Engineering: Theory, Methods and Applications*, **4**:317–367, 1998.

[41] M Koot, J T B A Kessels, B De Jager, W P M H Heemels, P P J Van Den Bosch, and M Steinbuch. Energy management strategies for vehicular electric power systems. *IEEE Transactions on Vehicular Technology*, **54**[3]:771–782, 2005.

[42] M. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, **32**[10]:1361–1379, October 1996.

[43] A Leonessa, M.M. Haddad, and V. Chellaboina. Nonlinear system stabilization via hierarchical switching control. *Automatic Control, IEEE Transactions on*, **46**[1]:17–28, Jan 2001.

[44] Liang Li, Wei Dong, Yindong Ji, Zengke Zhang, and Lang Tong. Minimal-energy driving strategy for high-speed electric train with hybrid system model. *Intelligent Transportation Systems, IEEE Transactions on*, **14**[4]:1642–1653, Dec 2013.

[45] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, **44**:2382–2387, 2008.

[46] D. Limon Marruedo, T. Alamo, and E.F. Camacho. Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC. In *American Control Conference, 2002. Proceedings of the 2002*, **1**, pages 364–369 vol.1, 2002.

[47] B. Lincoln and B. Bernhardsson. LQR optimization of linear system switching. *Automatic Control, IEEE Transactions on*, **47**[10]:1701–1705, Oct 2002.

[48] L. Liung. MATLAB® System Identification™ Toolbox, Getting Started Guide, 2014.

[49] Yaohui Lu and Yaman Arkun. Quasi-min-max MPC algorithms for LPV systems. *Automatica*, **36**[4]:527 – 540, 2000.

[50] T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. MPC for a low consumption electric vehicle with time-varying constraints. In *5th IFAC Symposium on System, Structure and Control and 2013 IFAC Joint Conference SSSC, FDA, TDS*, Grenoble, France, 2013.

[51] D.Q. Mayne, S.V. Rakovi, R. Findeisen, and F. Allgwer. Robust output feedback model predictive control of constrained linear systems: Time varying case. *Automatica*, **45**[9]:2082 – 2087, 2009.

[52] D.Q. Mayne, S.V. Raković, R. Findeisen, and F. Allgwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, **42**[7]:1217 – 1222, 2006.

[53] D.Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, **36**:789–814, 2000.

[54] D.Q. Mayne, M.M. Seron, and S.V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, **41**[2]:219 – 224, 2005.

[55] D.L. Pepyne and C.G. Cassandras. Optimal control of hybrid systems in manufacturing. *Proceedings of the IEEE*, **88**[7]:1108–1123, July 2000.

[56] B. Piccoli. Necessary conditions for hybrid optimization. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, **1**, pages 410–415 vol.1, 1999.

[57] Sabine Piller, Marion Perrin, and Andreas Jossen. Methods for state-of-charge determination and their applications. *Journal of Power Sources*, **96**[1]:113 – 120, 2001.

[58] A. V. Rao, D. A. Benson, C. L. Darby, M. A. M. A. Patterson, C. Francolin, I. Sanders, and G. T. Huntington. Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method. *ACM Transactions on Mathematical Software*, **37**[2], April–June 2010.

[59] P. Riedinger, FR. Kratz, C. Iung, and C. Zanne. Linear quadratic optimization for hybrid systems. *IEEE Conference on Decision and Control*, **3**:3059–3064, 1999.

[60] J.J. Santin. *The world's most fuel efficient vehicle: design and development of PAC-Car II.* vdf Hochschulverlag AG an der ETH Zürich, 2007.

[61] A. Sciarretta and L. Guzzella. Control of hybrid electric vehicles. *IEEE Control Systems Magazine*, **27**[2]:60–70, 2007.

[62] P. O. M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *Transactions on automatic control*, **43**:1163–1169, 1998.

[63] P.O.M. Scokaert, J.B. Rawlings, and E.S. Meadows. Discrete-time stability with perturbations: application to model predictive control. *Automatica*, **33**[3]:463 – 470, 1997.

[64] Zhendong Sun and S.S. Ge. Analysis and synthesis of switched linear control systems. *Automatica*, **41**[2]:181 – 195, 2005.

[65] H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, **1**, pages 425–430 vol.1, 1999.

[66] N. Wada, K. Saito, and M. Saeki. Model Predictive Control for Linear Parameter Varying Systems Using Parameter Dependent Lyapunov Function. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, **53**[12]:1446–1450, Dec 2006.

[67] N. Wada, H. Tomosugi, and M. Saeki. Model predictive tracking control for a linear system under time-varying input constraints. *International Journal of Robust and Nonlinear Control*, **23**[9]:945–964, 2013.

[68] Z. Wan and M. V. Kothare. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, **39**[5]:837 – 846, 2003.

[69] Liuping Wang. *Model predictive control system design and implementation using MATLAB*. Advances in industrial control. Springer, London, cop. 2009.

[70] Xuping Xu and P.J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *Automatic Control, IEEE Transactions on*, **49**[1]:2–16, Jan 2004.

[71] Shuyou Yu, C. Bohm, Hong Chen, and F. Allgower. MPC with one free control action for constrained LPV systems. In *Control Applications (CCA), 2010 IEEE International Conference on*, pages 1343–1348, Sept 2010.

Abstract

In the field of transportation, the research on energy efficiency has been carried out for few decades by the automotive industry, where one of the main objectives is to reduce the energetic consumption. This particular problem can be rephrased as how the vehicle must be driven so that the minimum quantity of energy is used. This is the optimal driving strategy. In this project, a suitable model of the Vir'volt electric vehicle involved in the European Shell Eco-Marathon is obtained. The unknown parameters involved in the vehicle dynamics are estimated using Parameter identification from experimental data. The identified dynamics is used to derive an optimal driving strategy that is intended to be tracked on-line during the driving task. The tracking task is subject to time-varying polytopic constraint on the input and/or the state. A MPC-based tracking strategy that uses an homothetic transformation as a suitable time-varying invariant set is used. The time-varying invariant set guarantees the asymptotic stability of the control law. The problem of the MPC tracking for Linear Parametric Varying (LPV) systems is introduced. A new explicit MPC strategy for LPV systems is developed. This strategy uses a Parameter dependent Lyapunov Function (PDLF) to involve explicitly the time-varying parameter in the control law and so it reduces conservatism. A benchmark is used to test the performances of the optimal driving strategy and the explicit MPC tracking strategy. Finally, a robust adaptive technique with on-line identification of the dynamics is has been proposed and tested in the race showing good performances of the adaptive driving strategy.

Keywords: Real-time applications, Optimal control, Shell Eco-Marathon, Electric vehicle, Model Predictive Control (MPC), Invariant sets, Time-varying invariant set, Homothetic invariant set, Linear Parametric Varying (LPV), Parameter dependent Lyapunov Function (PDLF), Adaptive control, Robust control.

Résumé

Le problème de l'efficacité énergétique dans le domaine des transports a comme principal défi savoir comment utiliser la source d'énergie pour que l'efficacité énergétique puisse être maximisée, c'est-à-dire comment le véhicule doit être conduit de telle sorte que la quantité minimale d ?énergie est utilisée. Ce problème est le principal problème considéré dans cette thèse. Le véhicule est un prototype impliqué dans la course européenne Shell Eco-Marathon. La dynamique du véhicule est d'abord obtenu par l'identification expérimentale des paramètres. Une stratégie en boucle ouverte de conduite optimale en termes de consommation électrique est calculée. Plusieurs approches ont été étudiées pour le suivi de la référence optimale (stratégie de conduite optimale). Ces approches doivent prendre en compte les ressources limitées en taille mémoire et capacité de calcul. Une commande prédictive (MPC) basée sur la dynamique linéarisée est tout d'abord synthétisée. Le problème de poursuite nécessite une MPC avec contraintes variant dans le temps. La stabilité et la convergence de la commande prédictive sont prouvées à l'aide du formalisme des ensembles invariants. En troisième partie, à partie du modèle LPV, une adaptation de techniques standards basées sur des fonctions de Lyapunov quadratiques et à paramètres variants avec calculs hors-ligne est proposée. Elle est implémentée sur un banc de test. Enfin, une technique adaptative robuste avec identification en ligne de la dynamique est proposée et implémentée dans le véhicule. Cette technique a été testée et validée en course. Les résultats expérimentaux obtenus montrent de bonnes performances de la stratégie de conduite.

Mots-clés: Contraintes temps réel, Commande optimale, Shell Eco-Marathon, Véhicule electrique, Commande prédictive (MPC), Ensembles invariants, Ensemble invariant homothétique, Linéaire à Paramètres Variant dans le temps (LPV), Fonction de Lyapunov à paramètres variants (PDLF), Commande adaptative, Commande robuste.