



HAL
open science

Contributions à la Commande Prédictive Non Linéaire pour les Systèmes à Dynamiques Rapides

André Murilo

► **To cite this version:**

André Murilo. Contributions à la Commande Prédictive Non Linéaire pour les Systèmes à Dynamiques Rapides. Automatique / Robotique. Institut National Polytechnique de Grenoble - INPG, 2009. Français. NNT: . tel-00442847

HAL Id: tel-00442847

<https://theses.hal.science/tel-00442847>

Submitted on 22 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

□□□□□□□□□□

THESE

pour obtenir le grade de

DOCTEUR DE L'Institut polytechnique de Grenoble

Spécialité : Automatique – Productique

préparée au laboratoire : GIPSA- Lab

dans le cadre de **l'Ecole Doctorale : Electronique, Electrotechnique,
Automatique, Télécommunications et Signal**

présentée et soutenue publiquement

par

André MURILO DE ALMEIDA PINTO

Le 02 décembre 2009

Titre :

**CONTRIBUTIONS A LA COMMANDE PREDICTIVE NON LINEAIRE
POUR LES SYSTEMES A DYNAMIQUES RAPIDES**

Directeur de thèse : Mazen Alamir

JURY

M. Jean-Pierre CORRIOU	(Professeur à l'INP-Lorraine)	, Président
M. Rolf FINDEISEN	(Professeur à l'Université de Magdeburg)	, Rapporteur
M. Didier DUMUR	(Professeur à Supelec)	, Rapporteur
M. Olivier SENAME	(Professeur à Grenoble INP)	, Examineur
M. Mazen ALAMIR	(Directeur de recherche CNRS)	, Directeur de thèse

À mes parents

À mon épouse

Remerciements

Je tiens tout d'abord à exprimer mes profonds remerciements à mon directeur de thèse Mazen Alamir. Ses conseils enrichissants, sa présence constante et surtout son côté humain ont été fondamentaux pour la réussite de cette thèse. Je serai éternellement reconnaissant pour l'aide compétente et l'encouragement qu'il m'a apportée pendant la thèse.

Je dois adresser aussi mes sincères remerciements à Jean-Pierre Corriou pour avoir accepté de présider le jury ainsi que Rolf Findeisen et Didier Dumur pour avoir bien voulu rapporter ce travail et Olivier Sename, pour avoir accepté le poste d'examineur.

La réalisation de cette thèse n'aurait pas été possible sans le soutien financier de la CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior) et le Ministère de l'Éducation brésilien. Je remercie donc ces deux institutions qui m'ont permis d'effectuer cette thèse dans les meilleures conditions.

Je remercie également Peter Ortner, Richard Fürhapter et le professeur Luigi del Re, avec lesquels j'ai eu le plaisir de travailler à l'Université de Linz, pendant mon séjour en Autriche pour la validation expérimentale sur le banc d'essais du moteur diesel.

Je tiens aussi à remercier tout le personnel du Gipsa-Lab, notamment le département Automatique, pour leur aide, leur gentillesse et pour avoir pu travailler dans des conditions très agréables. Je remercie en particulier Marie-Thérèse, Patricia, Virginie, Daniel, Didier, Thierry et Jonathan. Je remercie également Christian Commault, directeur de l'École doctorale, avec qui j'ai eu le premier contact en 2004 et qui m'a ouvert les portes du laboratoire.

Je remercie spécialement Gabriel Buche et Pascal Bellemain pour leur amitié et la complicité partagée tout au long de ces trois ans. Je garderai avec moi de très bons souvenirs de notre travail au bâtiment E : les pendules jumeaux, le support en langage C, la cafétéria, les discussions interminables sur plusieurs sujets, les démos, entre autres.

Je remercie également Antoine Picot, mon collègue de bureau, pour tous les bons moments partagés au sein du laboratoire et, principalement, dans différents bars de Grenoble. Un grand merci à Sylvain Blainvillain, Monsieur le président du GAG, pour son amitié et sa gentillesse.

Je dois aussi remercier toutes les personnes que j'ai eu le plaisir de rencontrer à Grenoble, en particulier, Benoît, Sophie M., Ion, Nicolas, Kelit, Michael, Sandra D., Sophie F., Julien, Jeanne et Rafael. Je me souviendrai longtemps de toutes les soirées, sorties, repas et les excellents moments que nous avons passés ensemble. Un grand merci à Fabrice et Christophe, qui m'ont énormément aidé pendant mon séjour en France et qui étaient toujours là, pendant les bons et mauvais moments, pour me soutenir, m'encourager et bien sûr, pour faire la fête.

Comment ne pas remercier toutes les personnes qui ont gentiment traversé l'Atlantique pour nous rendre visite et pour apporter un peu du soleil brésilien : Isabella, João, Aline, Rodrigo H., Thais, Serginho, Tia Angela, Graça, Mãe, Pai, João Marcos, Claudia, Tatiana, Juninho, Flavia, Marcio, Rodrigo, Adriana, Dudu, Leo, Rose, Fabricio, Fernanda, Moacir et Juliana. Votre présence en France a été primordiale pour me donner encore plus de motivation et la sérénité nécessaire pour bien accomplir mes travaux de thèse.

Mes remerciements s'adressent aussi à toute ma famille au Brésil, notamment mes parents, mes frères et ma soeur pour leur aide et leur encouragement.

Et finalement, je tiens à remercier mon épouse Sandra, qui a toujours été à mes côtés pendant ce long et laborieux parcours, et sans laquelle cette aventure n'aurait jamais été possible. Je la remercie de tout mon coeur.

Contents

Résumé Général	1
1 Introduction	23
1.1 General Aspects of Model Predictive Control	23
1.2 Motivations	26
1.3 Outline of the Dissertation	26
2 Nonlinear Model Predictive Control	29
2.1 Introduction	29
2.2 Historical Background	30
2.3 Problem Formulation	31
2.3.1 Some Definitions and Notation	31
2.3.2 The Cost Function	33
2.3.3 The NMPC State Feedback	33
2.4 Stability Analysis	34
2.5 Real-Time Issue	38
2.5.1 Distributing the Optimization over the System Life-Time . . .	38
2.5.2 The Extended System	39
2.6 Conclusion	41
3 NMPC for Fast Systems	43
3.1 Introduction	43
3.2 Fast NMPC: State of the Art	43
3.2.1 Linear Formulations	44
3.2.2 Nonlinear Formulations	47
3.3 The Parameterized NMPC	55
3.3.1 Motivations	55
3.3.2 Definitions and Notation	56
3.4 Conclusion	58
4 NMPC for a Diesel Engine Air Path	61
4.1 Introduction	61
4.2 Diesel Engine	62
4.2.1 General Aspects	62

4.2.2	Formation of Emissions	65
4.2.3	The Control Problem	66
4.2.4	Existing Control Strategies	68
4.2.5	System Modeling	70
4.3	Observer Design	79
4.3.1	The Relevance of the Observer	79
4.3.2	Moving Horizon Observer Design	80
4.3.3	Simulation Results	82
4.4	Control Design	83
4.4.1	Introduction	83
4.4.2	Some Considerations About the Engine's Behavior	84
4.4.3	The Steady Solution	86
4.4.4	Parametric NMPC Formulation for Diesel Engines	89
4.4.5	A Black-Box NMPC Solution for Diesel Engines	93
4.5	Optimization Process	95
4.5.1	Low Dimensional Optimization	95
4.5.2	A New Mixed SQP/Gradient/Trust Region Method	99
4.6	Simulation Results	104
4.6.1	Simulations Using the Model Depending on the Outputs	104
4.6.2	Simulations Using the Model Depending on the Inputs	107
4.6.3	Simulations Using the Full Nonlinear Model	110
4.7	Experimental Results	118
4.7.1	Successive Steps Sequence	119
4.7.2	NEDC Sequence	121
4.8	Conclusion	123
5	NMPC for a Twin-Pendulum System	125
5.1	Introduction	125
5.2	The Inverted Pendulum Problematic	126
5.2.1	Background on the Existing Variations of the Problem	126
5.2.2	The Twin-Pendulum System	128
5.3	System Modeling	130
5.3.1	The Model of the Cart/Pendulum System	131
5.3.2	The Model of the DC Motor	132
5.3.3	The Complete Model	133
5.3.4	The Control Problem	134
5.4	Parameterized NMPC	134
5.4.1	Model Analysis	134
5.4.2	The Control Design	135
5.4.3	The Hybrid Controller	141
5.4.4	Simulation Results	143
5.5	Observer Design	148
5.5.1	Introduction	148

5.5.2	Nonlinear Observer Design	149
5.5.3	Convergence Analysis	151
5.5.4	Simulation Results	153
5.6	Real-Time Implementation	155
5.6.1	Identification Process	155
5.6.2	Dealing with Friction	157
5.7	Experimental Results	160
5.8	Conclusion	165
Conclusion and Future Work		167
A Moving Horizon Observer Matrices		171
B Diesel Engine: Technical Data		173
B.1	Test Facilities	173
B.2	Diesel Engine Parameters	175
B.3	Block Diagram	176
C Twin-Pendulum System: Technical Data		177
C.1	Test Facilities	177
C.2	Twin-Pendulum System Parameters	182
C.3	Block Diagram	183
Bibliography		184

List of Figures

1.1	Schematic view of the MPC strategy	24
3.1	Schematic view of the partitioning of the state space in the explicit approach	45
3.2	Schematic view of two scenarios concerning the homotopy paths . . .	47
3.3	Schematic view of the multiple shooting strategy applied to NLP problems	49
3.4	Schematic view of the preparation and the feedback phases present in the real-time iteration method	50
3.5	Schematic view showing how an approximate but well posed optimization problem may be worth using than an exact but highly accidental optimization problem.	56
4.1	BMW Diesel engine.	62
4.2	Diesel car market share in western Europe between 1990 and 2008. . .	63
4.3	Schematic view of the temperature and chemistry of a typical diesel engine flame	65
4.4	Schematic view of the trade-off between NOx and PM for diesel vehicle. . .	66
4.5	Schematic view of the complete diesel engine air path	67
4.6	Polynomial map $T_{xi} = \mathcal{P}_1(T_x, W_{xi})$	71
4.7	Schematic view of the simplified diesel engine air path and the variables used for control design.	76
4.8	Simulation results related to the state estimation error.	82
4.9	Simulation results showing the evolution of the offset error estimation. . .	83
4.10	Typical behavior of the mass air flow showing a non-minimum phase property.	84
4.11	Open-loop behavior at $N_e=1800$ RPM and $w_f=20$ mg/cyl under a step sequence of EGR and VGT	85
4.12	Schematic view of the feasible set-points, at $w = (0, 0)$	88
4.13	Evolution of u^*	89
4.14	Notations related to the moving horizon observer and the predictive control	92
4.15	Simulation results showing the evolution of p_1 et p_2	97
4.16	Simulation results showing the measured values of MAF and MAP . . .	97

4.17	The shape of the cost function $J(\cdot, p, \hat{x}(t_i))$ for the three particular instants depicted on Figures 4.15 and 4.16	98
4.18	Schematic view of the preparation phase of the scalar SQP routine	99
4.19	The SQP scalar algorithm performs successively scalar SQP's on each component i , each followed by an adaptation of the corresponding trust region	100
4.20	Situation when the test $J_{cand} < J_{act}$ fails	101
4.21	Evolution of the closed loop system with a sequence of filtered steps (Model (4.6)).	105
4.22	Evolution of the closed loop system with a variable signal reference (Model (4.6)).	106
4.23	Histograms of the computation time needed at at each sampling period to perform the observation task and the $n_{fe} = 10$ function evaluations of the optimization procedure (Model (4.6)).	106
4.24	Evolution of the closed loop system with the introduction of measured disturbances, engine speed and fuel injection (Model (4.6)).	107
4.25	Evolution of the measurement disturbances profile (w) used to generate MAF and MAP set-points (y) (Model (4.7)).	108
4.26	Evolution of the closed loop system under the successive steps sequence on MAF and MAP (Model (4.7)).	109
4.27	Evolution of the set of parameters and the steady control for the scenario depicted on Figure 4.26 (Model (4.7)).	109
4.28	Evolution of the closed-loop output under a sequence of step changes and $t_r = 0.1s$	112
4.29	Evolution of the closed-loop output under a sequence of step changes and $t_r = 1s$ (Model (4.3)-(4.5)).	112
4.30	Evolution of the offset error estimation and dynamics error between the predicted and measured value (Model (4.3)-(4.5)).	113
4.31	Evolution of the 4 parameters used for the optimization problem (4.32), which considers the stationary inputs as a part of the decision variable (Model (4.3)-(4.5)).	114
4.32	Evolution of the computation time of the parameterized NMPC scheme	115
4.33	Simulation results showing the evolution of MAF and MAP over the whole NEDC tracking, with $\rho_2 = 1$	116
4.34	Evolution of MAF and MAP over the NEDC tracking, with $\rho_2 = 100$ (on the top) and $\rho_2 = 0.01$ (on the bottom) (Model (4.3)-(4.5)).	117
4.35	The profiles of fuel injection and speed engine used for the set-point generation.	118
4.36	ECU's response for a step sequence	119
4.37	Experimental results of the parameterized NMPC scheme on a sequence of steps on the set-points	120
4.38	Experimental results of the parameterized NMPC scheme, with $\rho_x = 1000$	121

4.39	Experimental results comparing the ECU and the parameterized NMPC scheme under the extra-urban part of the NEDC	122
5.1	Classical single inverted pendulum problem	126
5.2	Schematic view of a double-pendulum system	127
5.3	Representation of the Furuta rotary inverted pendulum.	127
5.4	Twin-pendulum system of Control Systems Department (University of Grenoble) used as experimental platform.	128
5.5	Rotary twin-pendulum without constraints on the angular position of the steering devices.	129
5.6	Twin-pendulum system (Control Systems Department, Grenoble). . .	130
5.7	Schematic view of the predicted evolution of E over the prediction horizon when $T_p \geq t_r$	137
5.8	Definition of the allowable increment d_r of the cart position as a function of the energy level E	139
5.9	Definition of the admissible range for the swing-up phase and the margins for the linear phase	140
5.10	Intuitive view showing the contraction of E according to the optimum values of \hat{p}	141
5.11	The transient of the closed-loop trajectory under the linear controller (5.33) may not respect the constraint on the cart position.	142
5.12	Evolution of the closed-loop system under $\delta_r^{max} = 0.4m$ (Table 5.1). .	145
5.13	Evolution of the closed-loop system under $\delta_r^{max} = 0.1m$ (Table 5.1). .	145
5.14	Evolution of the closed-loop system under $t_r = 0.1s$ (Table 5.2). . .	146
5.15	Evolution of the closed-loop system under $t_r = T_p = 0.6s$ (Table 5.2). .	147
5.16	Evolution of the closed-loop system starting from the initial state $x_0 = [0, 0, 120 \text{ deg}, 1.5 \text{ rad/s}, -100 \text{ deg}, -2 \text{ rad/s}]$ (Table 5.2).	147
5.17	Closed-loop system behavior with a bias of 1 <i>deg</i> on both angular sensors	148
5.18	Evolution of θ_1, θ_2, r , the estimation of the offset error $[\varepsilon_1, \varepsilon_2] = [-2, 3] \text{ deg}$ and the norm of the estimation error.	154
5.19	Evolution of the twin-pendulum system behavior under the presence of a variable bias $[\varepsilon_1, \varepsilon_2] = [3 + 0.06 \sin(t), 2 = 5 - 0.06 \sin(t)] \text{ deg}$ on the angular sensors.	154
5.20	Open-loop experiments: the cart is fixed and the pendulums oscillate freely around the downward equilibrium.	155
5.21	Real and identified data using the FMINCON routine.	156
5.22	The computation of the optimal set-point r_d is done by checking the values of the cost function for n_{fe} uniformly distributed grid points in the interval $[r(k) - d_r(E), r(k) + d_r(E)]$	158
5.23	Schematic view of the hysteresis introduced for the switch condition. .	159
5.24	Graphical User Interface (GUI) for the twin-pendulum system.	161

5.25	Experimental results showing the efficiency of the hybrid control strategy based on the parameterized contractive NMPC scheme in stabilizing the twin-pendulum system.	162
5.26	Second scenario starting from a different initial condition on the angular positions and speeds. Note the important role of the virtual constraint r_{max}^{RHC} at $t \in [9, 15]s$	163
5.27	Last experiment showing the evolution of the twin-pendulum system under the parameterized NMPC scheme.	163
5.28	Evolution of the torque of the DC motor for the preceding scenarios.	164
B.1	Block diagram used for simulation.	176
C.1	Block diagram for the cart speed signal	177
C.2	Block diagram for the angular position signal	178
C.3	System architecture of the twin-pendulum system.	179
C.4	Complete description of the DC motor	180
C.5	Anti-aliasing filter used for the cart speed	181
C.6	Block diagram used in real-time implementation	183

List of Tables

3.1	Some computation time examples for the successful use of parameterized NMPC in robotics and mechatronic	57
4.1	Sections where the models are referred	79
4.2	Parameters of the solver used in simulations	104
4.3	Parameters used in simulations - Model (4.6)	105
4.4	Parameters used in simulations - Model (4.7)	108
4.5	Parameters used in simulations - Model (4.3)-(4.5)	111
4.6	Parameters used in experiments - Model (4.7)	119
5.1	Parameters used in the first part of simulations	144
5.2	Parameters used in the second part of simulations	144
5.3	Parameters used in experiments	157
5.4	Parameters used in experiments	160
B.1	Board Description	174
B.2	Full Nonlinear Model Parameters (4.3)-(4.5)	175
B.3	Empirical Model Parameters (4.6) and (4.7)	175
C.1	Interface Boards	179
C.2	DC Motor Parameters	182
C.3	Physical Parameters: Pendulums and Cart	182

Acronyms

MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
RHC	Receding Horizon Control
CLF	Control Lyapunov Function
IDCOM	Identification and Command
DMC	Dynamic Matrix Control
MPHC	Model Predictive Heuristic Control
QDMC	Quadratic Dynamic Matrix Control
QP	Quadratic Program
SQP	Sequential Quadratic Program
IPOPT	Interior Point Optimizer
MUSCOD	Multiple Shooting Code for Optimization
PWA	Piece-Wise Affine
PWC	Piece-Wise Constant
CR	Critical Region
C/GMRES	Continuation/Generalized Minimum Residual
ODE	Ordinary Differential Equation
DAE	Differential Algebraic Equation
LP	Linear Programming
NLP	Nonlinear Programming
KKT	Karush-Kuhn-Trucker
CI	Compression-Ignition
SI	Spark-Ignition
NO_x	Oxides of Nitrogen
PM	Particulate Matter
uHC	Unburnt Hydrocarbons
MAF	Mass Air Flow
MAP	Manifold Absolute Pressure
EGR	Exhaust Gas Recirculation
VGT	Variable Geometry Turbine
LPV	Linear Parameter Varying
MVM	Mean Value Model
MHO	Moving Horizon Observer
EKF	Extended Kalman Filter
NEDC	New European Driving Cycle
ECU	Engine Control Unit
LQR	Linear Quadratic Regulator
GUI	Graphical User Interface

Résumé Général

La Commande Prédictive Non Linéaire

La commande prédictive est une méthodologie avancée de commande qui utilise explicitement le modèle du système pour prédire les évolutions futures du procédé. L'idée de base de la commande prédictive consiste à calculer, à chaque instant d'échantillonnage, une séquence de commande optimale sur un horizon de prédiction visant à minimiser une certaine fonction coût qui exprime l'objectif de la commande. Toutefois, seule la première valeur de cette séquence est appliquée au système à l'instant d'échantillonnage suivant. Ce processus d'optimisation est répété à chaque instant d'échantillonnage et par conséquent, l'horizon de prédiction est successivement décalé à l'avant. Pour cette raison, la commande prédictive est aussi appelée de *commande à l'horizon fuyant*. La figure 1 illustre la stratégie générale de la commande prédictive.

Initialement conçue pour répondre aux besoins des procédés dits *lents*, notamment l'industrie pétrochimique, la commande prédictive a rapidement trouvé sa place dans d'autres domaines, plus particulièrement pour les systèmes non linéaires, c-à-d la *Commande Prédictive Non Linéaire* (CPNL). La CPNL peut être attractive pour les raisons suivantes [Mayne et al., 2000]:

- La gestion des contraintes
La prise en compte des contraintes économiques, de sécurité et sur les actionneurs sont inévitables dans plusieurs domaines d'application. La présence de ces contraintes fait que la CPNL soit, dans la plupart des cas, la seule option pour le projet des contrôleurs.
 - La prise en compte des non linéarités
Il existe plusieurs situations où les non linéarités ne sont pas négligeables et les modèles linéaires ne peuvent pas représenter le vrai procédé correctement. La CPNL est basée sur l'indépendance vis-à-vis de la structure mathématique
-

du modèle, et par conséquent, elle peut être appliquée à une gamme plus large des procédés industrielles complexes.

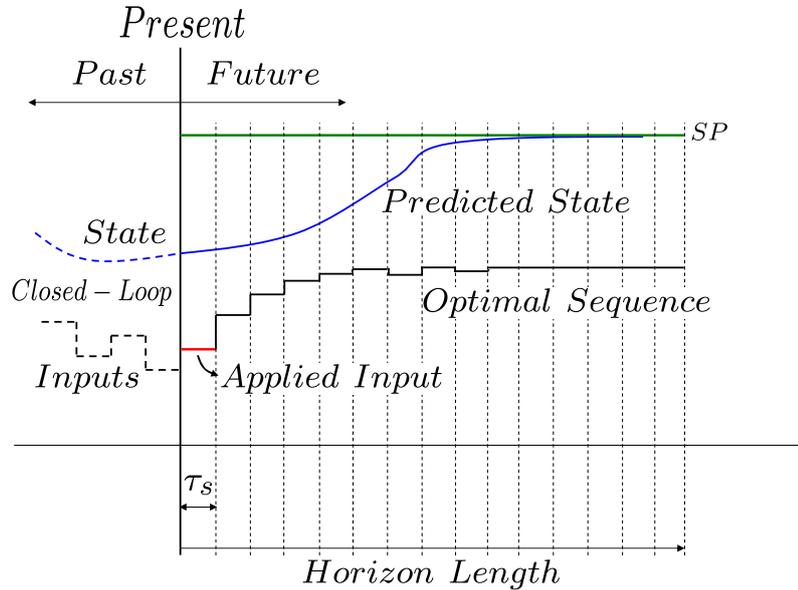


Figure 1: La stratégie de la commande prédictive

Les avantages cités ci-dessus font de la CPNL un outil très puissant pour le projet des contrôleurs. D'autre part, quelques inconvénients sont à considérer tels que:

- **Robustesse**
 Pour l'instant, en ce qui concerne la commande prédictive, il n'y a pas encore une réponse satisfaisante pour résoudre ce problème, malgré l'énorme effort de la part de la communauté scientifique [Magni and R.Scattolini, 2007]. La difficulté majeure consiste à gérer les incertitudes dans les cas où des grands horizons de prédiction sont utilisés, et ceci est souvent le cas. Une bonne option pour attaquer le problème de la robustesse consiste à coupler le contrôleur prédictif à un correcteur robuste local dans une boucle interne.
- **Implémentation temps réel**
 En effet, la résolution en ligne de problèmes d'optimisation généralement non convexes et contraints devient un problème lorsque le temps disponible pour le calcul n'est qu'une fraction de la période d'échantillonnage. Si la réalisation de cette tâche est compatible avec les temps caractéristiques des procédés chimiques par exemple, elle l'est beaucoup moins pour les systèmes à dynamiques rapides comme ceux de la robotique ou de la mécatronique. Dans ce cas, la formulation du problème d'optimisation doit tenir compte explicitement des contraintes de mise en oeuvre en temps réel. Cette prise de conscience se cristallise depuis une décennie seulement, et le principe de base est montré dans la section suivante.

La CPNL pour les Systèmes Rapides

D'une façon générale, un système peut être considéré comme rapide si le temps de réponse associé à ce système est plus petit que le temps nécessaire pour résoudre un problème d'optimisation classique en boucle ouverte. Nous pouvons diviser les approches dans deux groupes principaux: les méthodes linéaires et non linéaires. Parmi les approches linéaires, deux méthodes méritent une attention spéciale:

- Approche explicite [Bemporad et al., 2002]
Le principe principal de cette approche s'appuie sur la division de l'espace d'état en plusieurs régions critiques, dans laquelle le problème d'optimisation résultant peut être obtenu explicitement. Pour ce faire, l'algorithme tourne hors-ligne pour définir toutes ces régions. Ceci peut réduire considérablement le temps de calcul, car la seule tâche temps-réel à exécuter consiste à choisir la région correspondante selon l'état actuel. D'autre part, ces calculs hors-ligne peuvent augmenter de façon exponentielle selon la dimension du système, ce qui peut être très problématique dû aux éventuelles limitations de mémoire et de stockage.
- Approche basée sur les ensembles actifs [Ferreau et al., 2006]
Cette approche vise à éviter les calculs hors-ligne de la méthode précédente, à travers un processus temps-réel pour l'obtention de la région critique. Dans ce cas, la trajectoire de l'état est gérée par un paramètre de continuation $\tau \in [0, 1]$, ce qui fait les transitions parmi les différentes régions critiques. Ce principe est utilisé par le logiciel *qpOASES* [Ferreau, 2007] et a été validé sur quelques applications pratiques [Ferreau et al., 2007].

Néanmoins, les approches présentées ci-dessus ne concernent que les systèmes linéaires. En fait, le succès des approches prédictives linéaires est principalement dû au fait que la solution complète du problème original d'optimisation non linéaire serait, de toute façon, infaisable, étant donné la limitation du temps de calcul. Cette difficulté, considérée comme insurmontable pendant longtemps, a inspiré l'idée de distribuer le processus d'optimisation sur le temps réel du procédé. Quelques contributions importantes ont été développées et sont brièvement décrites ci-dessous.

- Approche par tirs multiples [Diehl et al., 2005a]
Dans cette approche, le problème d'optimisation à la base de la formulation prédictive est transformé en un problème non linéaire où la variable de décision est définie non seulement par la commande, mais aussi par les états et les multiplicateurs de Lagrange associés aux contraintes. Le principal avantage de cette méthode vient du fait que la technique de tirs multiples permet
-

d'éviter les problèmes d'instabilité qui pourraient surgir suite à une mauvaise initialisation de la trajectoire de commande, ce qui peut être très récurrent dans les méthodes directes (variables de décision constantes par morceaux).

- Méthode des points intérieurs [Biegler and Zavala, 2009]
Cette méthode a été conçue pour les systèmes industriels à très grande taille, avec des nombreuses variables d'états et équations différentielles. La méthode de points intérieurs est utilisée pour résoudre le problème d'optimisation non linéaire d'une façon très efficace [Zavala et al., 2006]. Ainsi que pour l'approche par tirs multiples, cette méthode utilise aussi l'état comme variable de décision ce qui augmente considérablement la taille du système à résoudre. L'algorithme *IPOPT* [Wächter and Biegler, 2006] est basé sur cette méthode et a été appliqué avec succès dans quelques procédés industrielles à grande taille.
- Approche par continuation [Ohtsuka, 2004]
L'approche par continuation, aussi connue par le nom *Continuation/Generalized Minimum RESidual (C/GMRES)*, utilise comme variable de décision la séquence de commande et les multiplicateurs de Lagrange associés aux conditions d'optimalité. Une simple équation différentielle dépendante de ces variables est définie, dont la solution ne passe pas par la solution d'un problème d'optimisation mais par un retour d'état dynamique. Toutefois, l'approche par continuation reste toujours une méthode directe et donc, présente les mêmes risques d'intégrer une séquence instable lorsque des mauvaises séquences de commande sont trouvées durant le transitoire de la boucle fermée. Ceci incite à utiliser l'approche par tirs multiples comme une deuxième couche pour essayer de contourner ce problème. Il existe aussi l'approche présentée par [DeHaan and Guay, 2007] qui partage aussi la même idée que [Ohtsuka, 2004], à travers un schéma de mise-à-jour basée sur l'équation différentielle. Cependant, les instants de prise de décision font aussi partis de la variable de décision, ce qui représente la particularité de cette méthode.

D'autre part, *l'approche paramétrique* introduite par [Alamir, 2006a] a été utilisée dans cette thèse. Cette approche possède quelques avantages intéressants, à savoir:

- Tandis que les méthodes classiques utilisent la paramétrisation triviale où toute la séquence de commande fait partie de la variable de décision (et parfois même les variables d'état), l'approche paramétrique favorise l'obtention des problèmes d'optimisation à faible dimension. Ceci est particulièrement vrai surtout pour les systèmes mécatroniques et robotiques.
 - L'approche paramétrique favorise l'obtention de problèmes à faible dimension et bien posés vis-à-vis le processus d'optimisation. Autrement dit, la solution sous-optimale d'un problème d'optimisation simple peut être meilleure par rapport à celle issue d'un problème plus complexe.
-

- La simulation en boucle ouverte de séquences temporelles de commandes relativement simples, utilisées dans le contexte d'horizon fuyant, peuvent mener à des profils extrêmement riches en boucle fermé.

Ceci dit, il est nécessaire d'introduire quelques formalismes concernant l'approche paramétrique. Considérons le modèle général d'un système dynamique invariant dans le temps:

$$x(t) = X(t, x_0, \mathbf{u}) \quad x \in \mathbb{R}^n \quad (1)$$

où $x(t)$ est l'état du système à l'instant t partant des conditions initiales $x(0) = x_0$ et sous le profil de commande \mathbf{u} qui est défini sur $[0, T]$ à valeurs dans l'ensemble compact admissible $\mathbb{U} \subset \mathbb{R}^m$. Soit $\tau_s > 0$ une période d'échantillonnage et $T = N\tau_s$. Toute application $\mathcal{U}_{pwc} : \mathbb{P} \times \mathbb{R}^n \rightarrow \mathbb{U}^N$ défini sur $[0, T]$ un profil de commande admissible constant par morceaux à paramètres dans $\mathbb{P} \subset \mathbb{R}^{n_p}$ par:

$$\begin{aligned} \mathbf{u}(t) &= u^{(k)}(p, x) \quad ; \quad t \in [t_{k-1}, t_k] \quad ; \quad t_k = k\tau_s \\ \mathcal{U}_{pwc}(p, x) &:= (u^{(1)}(p, x) \quad \dots \quad u^{(N)}(p, x)) \in \mathbb{U}^N \end{aligned}$$

Par abus de notation, l'évolution de l'état sous le profil de commande $\mathcal{U}_{pwc}(p)$ sera noté $X(t, x_0, p)$. En réalité, à chaque instant d'échantillonnage $t_j = j\tau_s$ ($j \in \mathbb{N}$), $X(t, x(t_j), p)$ est la valeur de l'état à l'instant $t_j + t$ sous le profil de commande $\mathcal{U}_{pwc}(p, x)$ défini par le paramètre p . La commande prédictive consiste à résoudre à chaque instant d'échantillonnage t_j , le problème d'optimisation suivant:

$$\hat{p}(x(t_j)) := \arg \min_{p \in \mathbb{P}} [J(p, x(t_j))] \quad \text{sous} \quad C(p, x(t_j)) \leq 0 \quad (2)$$

et à appliquer la première commande de la séquence optimale $\mathcal{U}_{pwc}(\hat{p}, x(t_j))$, à savoir $K := u^{(1)}(\hat{p}(\cdot), \cdot)$ durant la période d'échantillonnage $[t_j, t_{j+1}]$. Noter que $J(x, p)$ représente une fonction coût portant sur l'évolution future sous la commande définie par p alors que $C(x, p)$ représente une contrainte pouvant regrouper des conditions à satisfaire à tout instant sur l'horizon de prédiction ainsi que des contraintes ponctuelles (finales en particulier). A l'instant d'échantillonnage suivant, le nouveau problème est formulé avec la nouvelle valeur de l'état $x(t_{j+1})$ et la nouvelle commande est appliquée durant l'intervalle $[t_{j+1}, t_{j+2}]$ et ainsi de suite. En fin de compte, la commande prédictive n'est que le retour d'état implicite:

$$K := u^{(1)}(\hat{p}(\cdot), \cdot) : \mathbb{R}^n \rightarrow \mathbb{U} \quad (3)$$

qui se traduit simplement par la recherche de la meilleure stratégie (une séquence d'actions) étant donné l'état actuel du système, commencer l'exécution de la première action de la séquence puis reconsidérer systématiquement la situation. Dans la suite, l'approche paramétrique est utilisée dans deux exemples d'application: le moteur diesel et le système des pendules jumeaux.

La CPNL paramétrique pour le Moteur Diesel

Le Moteur Diesel

L'intérêt porté sur le moteur diesel a considérablement augmenté ces dernières années. En effet, ce n'est pas par hasard que ce type de moteur attire de plus en plus l'attention du marché automobile. Le couple élevé à faible vitesse, l'efficacité, la consommation réduite et la durabilité sont les principaux avantages de ce genre de moteur. En revanche, les oxydes d'azote (NOx) et les particules issues de la combustion du diesel représentent sans doute l'inconvénient le plus important. Les détails du processus de combustion sont montrés dans [Heywood, 1988].

Afin d'attaquer le problème des émissions du moteur diesel, il faut d'abord comprendre comment fonctionne le circuit d'air d'un moteur diesel. Le circuit complet est illustré dans la figure 2. Le compresseur injecte l'air dans le moteur pour monter la pression. Le diesel pulvérisé est introduit dans la chambre de combustion et brûlé avec l'air injecté par le compresseur. Une partie du gaz sortant est réintroduit dans le moteur par le biais de la vanne de ré-circulation EGR (*Exhaust Gas Recirculation*), qui est très important pour réduire les NOx. La partie du gaz qui n'est pas ré-injectée est utilisée pour la vanne VGT (*Variable Geometry Turbine*), qui absorbe l'énergie du gaz pour propulser le compresseur et injecter l'air dans le système. Les vannes EGR et VGT sont donc les variables de commande u .

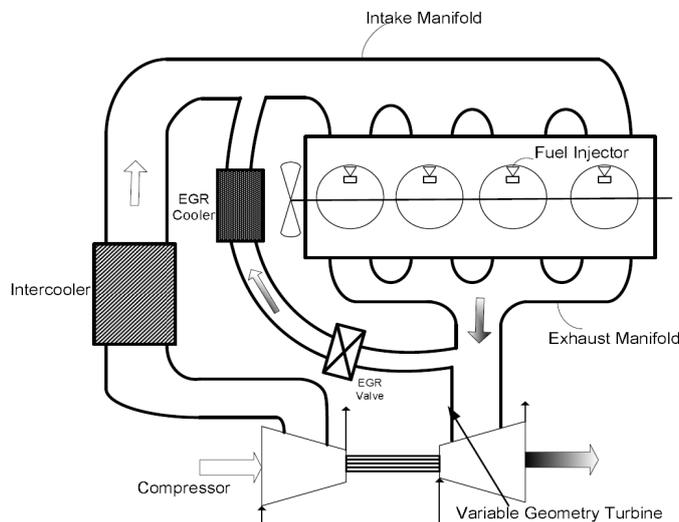


Figure 2: Le circuit d'air d'un moteur diesel [Wei, 2006].

Il faut souligner que, dans une voiture de série, il n'existe pas de capteur pour mesurer les émissions. Pour cette raison, le contrôle des émissions se fait à travers la poursuite des valeurs de la pression à l'entrée du moteur MAP (*Manifold Absolute*

Pressure) et le débit d'air MAF (*Mass Air Flow*), qui sont les sorties y du système. Par conséquent, la génération de set-points de MAP et MAF doit respecter les niveaux d'émissions. Cet aspect ne sera pas abordé dans cette thèse, et le lecteur peut se référer à [van Nieuwstadt et al., 2000] pour plus de détails à ce sujet. Par conséquent, l'objectif de la commande est de piloter les vannes EGR et VGT pour faire la poursuite de MAF et MAP selon les consignes désirés y^d . En plus, les deux entrées doivent respecter les contraintes suivantes:

$$u \in [u_{min}, u_{max}] \quad ; \quad u_{min} \in \mathbb{R}^2 \quad ; \quad u_{max} \in \mathbb{R}^2 \quad (4)$$

$$\delta u \in [-\delta_{max}, +\delta_{max}] \quad ; \quad \delta_{max} \in \mathbb{R}^2 \quad (5)$$

où δu définit l'incrément discret de l'entrée tel que $\delta u(k) = u(k+1) - u(k)$. Plusieurs approches ont été développées pour faire face à la poursuite des valeurs de MAF et MAP du moteur diesel [Jankovic and Kolmanovsky, 1998, Plianos and Stobart, 2007, Chauvin et al., 2006]. Plus récemment, quelques contributions utilisant la commande prédictive sont aussi apparues pour s'attaquer à ce problème. Dans [Ortner and del Re, 2007], l'approche explicite, présentée dans la section précédente, a été utilisée avec l'inconvénient relatif à la quantité de calculs hors-ligne pour générer toutes les régions critiques dans l'espace d'état. Pour éviter cela, l'approche basée sur les ensembles actifs a été utilisée dans [Ferreau et al., 2007] pour améliorer l'implementabilité temps-réel du contrôleur. En tout cas, il faut rappeler que ces approches se sont basées sur des modèles linéarisés au tour d'un point de fonctionnement du moteur, ce qui peut limiter énormément les performances et le champ d'action du contrôleur, vu que le moteur diesel est un système fortement non linéaire et contraint. Pour cette raison, la CPNL basée sur l'approche paramétrique est proposée pour réaliser la poursuite des valeurs de MAF et de MAP. L'avantage principal de cette stratégie est la caractéristique générique de la méthode, qui peut être utilisée avec n'importe quel type de modèle développé pour moteur diesel. Dans cette thèse, trois modèles non linéaires ont été utilisés pour montrer l'efficacité de cette méthode. Le premier est la représentation thermodynamique (6) de 6^{ième} ordre utilisée dans [Langthaler, 2007]

$$\begin{cases} x^+ = f(x, u, w) & ; \quad x \in \mathbb{R}^6 \\ y = g(x, u, w) \end{cases} \quad (6)$$

où $w \in \mathbb{R}^2$ sont les perturbations mesurées: la vitesse du moteur N_e et l'injection de carburant w_f . Les deux autres modèles ont été développés sur le banc d'essais à l'université de Linz, en Autriche, dont seulement un (7) est présenté dans ce résumé pour simplifier l'analyse, car tous les deux ont la même structure affine à l'état. Les résultats concernant l'autre modèle sont montrés dans [Murilo et al., 2009a].

$$\begin{cases} x^+ = [A(u - u_c, w - w_c)]x + B_1[u - u_c] + B_2[w - w_c] & ; \quad x \in \mathbb{R}^8 \\ y = [C(u - u_c, w - w_c)]x + y_c + \varepsilon \\ \varepsilon^+ = \varepsilon \end{cases} \quad (7)$$

où ε est l'erreur de prédiction et u_c , y_c et w_c sont les valeurs centrales autour desquelles le modèle a été identifié.

L'Observateur à Horizon Glissant

Contrairement au modèle physique (6) où tout le vecteur d'état est disponible pour les mesures, les modèles empiriques nécessitent d'un observateur pour reconstruire les états identifiés. Un observateur à horizon glissant MHO (*Moving Horizon Observer*) a été donc proposé pour ces deux modèles. Le MHO est le concept dual par rapport à la commande prédictive, et possède essentiellement les mêmes avantages et inconvénients [Michalska and Q.Mayne, 1995, Alamir, 1999]. Ce genre d'observateur peut gérer les non linéarités et les contraintes à travers la résolution d'un problème d'optimisation qui doit être résolu en respectant le temps de calcul disponible. L'idée de base d'un MHO consiste à estimer l'état actuel du système en utilisant une fenêtre glissante de taille N_o qui contient les valeurs passées. Les mesures plus anciennes sont abandonnées tandis que les plus récentes sont prises en considération. Puis, ces données sont utilisées pour définir une fonction coût dont la variable de décision est justement l'état désiré. Si nous considérons k comme l'instant actuel, \bar{Y}_k , \bar{U}_k et \bar{E}_k les vecteurs contenant des valeurs passées des mesures, entrées et l'erreur de prédiction respectivement, nous pouvons donc écrire:

$$\bar{Y}_k = \begin{pmatrix} y(k - N_o + 1) \\ \vdots \\ y(k) \end{pmatrix} \in \mathbb{R}^{m \cdot N_o}; \bar{U}_k = \begin{pmatrix} u(k - N_o + 1) \\ w(k - N_o + 1) \\ \vdots \\ u(k) \\ w(k) \end{pmatrix} \in \mathbb{R}^{2m \cdot N_o}; \bar{E}_k = \begin{pmatrix} \varepsilon(k - N_o + 1) \\ \vdots \\ \varepsilon(k) \end{pmatrix} \in \mathbb{R}^{m \cdot N_o}$$

La valeur de ε est mise-à-jour à chaque période d'échantillonnage selon $\varepsilon(k) = \varepsilon(k-1) + k_i \cdot (y^p(k) - y^m(k))$, où $y^p(k)$ et $y^m(k)$ sont respectivement la valeur prédite partant de l'état estimé à l'instant $k-1$ et la sortie à l'instant k , et $\varepsilon(k-1)$ la valeur précédente de l'erreur de prédiction et $k_i \in \mathbb{R}^m$ le gain d'intégration. Nous pouvons donc obtenir les relations suivantes:

$$x(k) = [\Phi(\bar{U}_k)] \cdot x^{(-)} + [\Psi(\bar{U}_k)] \cdot \bar{U}_k \quad (8)$$

$$\bar{Y}_k = [\Omega(\bar{U}_k)] \cdot x^{(-)} + [\Gamma(\bar{U}_k)] \cdot \bar{U}_k + \bar{E}_k \quad (9)$$

où $x^{(-)}$ est l'état à l'instant $k - N_o + 1$. Les matrices $\Phi(\bar{U}_k)$, $\Omega(\bar{U}_k)$, $\Gamma(\bar{U}_k)$ et $\Psi(\bar{U}_k)$ sont obtenues après quelques manipulations avec $A(\cdot)$, $C(\cdot)$ et $B = [B_1 \ B_2]$. Les détails de ces calculs sont montrés dans l'annexe A. Nous pouvons donc écrire les équations pour $\hat{x}(k-1)$ et $x^{(-)}$ tel que:

$$A(u(k-1), w(k-1))\hat{x}(k-1) + B[u(k-1) \ w(k-1)]^T = [\Phi(\bar{U}_k)] \cdot x^{(-)} + [\Psi(\bar{U}_k)] \cdot \bar{U}_k$$

$$\bar{Y}_k = [\Omega(\bar{U}_k)] \cdot x^{(-)} + [\Gamma(\bar{U}_k)] \cdot \bar{U}_k + \bar{E}_k$$

La première équation est liée à l'équation d'état et la deuxième au vecteur de sortie. Par conséquent, nous pouvons définir une fonction coût dont la variable de décision est l'état passé $x^{(-)}$ de la façon suivante:

$$\hat{x}^{(-)} = \underset{\xi \in \mathbb{R}^n}{\operatorname{argmin}} \|G_1 \cdot \xi - S_1\|_{Q_1}^2 + \|G_2 \cdot \xi - S_2\|_{Q_2}^2$$

où Q_1 et Q_2 sont les matrices de pondération qui peuvent être prises en tant que l'inverse des matrices de covariance, exactement comme pour le filtre de Kalman étendu, et les autres matrices sont définies comme suit:

$$G_1 = \Omega(\bar{U}_k) \quad ; \quad G_2 = \Phi(\bar{U}_k) \quad ; \quad S_1 = Y_k - \Gamma(\bar{U}_k) \cdot \bar{U}_k - \bar{E}_k$$

$$S_2 = A(u(k-1), w(k-1))\hat{x}(k-1) + B[u(k-1) \ w(k-1)]^T - \Psi(\bar{U}_k) \cdot \bar{U}_k$$

Avec les notations ci-dessus, l'équation de l'observateur peut être écrite de la façon suivante:

$$\hat{x}(k) = \Phi(\bar{U}_k) \cdot [(G_1^T \cdot Q_1 \cdot G_1 + G_2^T \cdot Q_2 \cdot G_2)^\dagger (S_1^T \cdot Q_1 \cdot G_1 + S_2^T \cdot Q_2 \cdot G_2)^T] \hat{x}_{k-1} + \Psi(\bar{U}_k) \cdot \bar{U}_k \quad (10)$$

La figure 3 montre l'évolution de l'erreur d'estimation de l'état, avec Q_1 et Q_2 étant les identités et $k_i = [3.5; 3.5]$. Un biais constant $\varepsilon = [-3, -15]$ a été introduit à $t = 0s$ et $\varepsilon = [-10, -5]$ à $t = 100s$. Malgré la présence de cette erreur, la convergence asymptotique de l'estimation de l'erreur est obtenue.

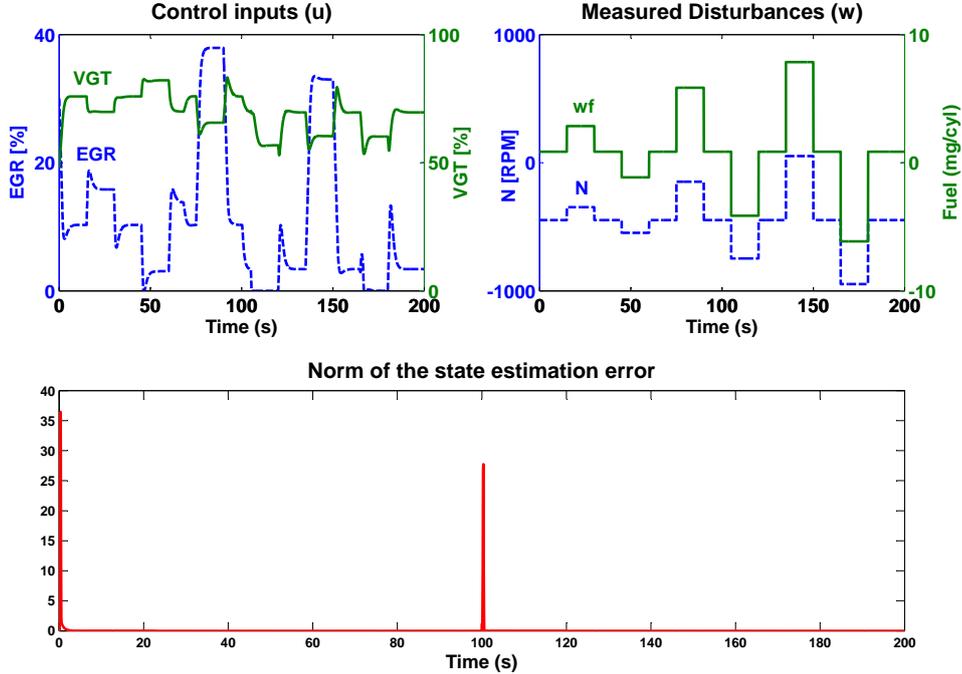


Figure 3: Résultats de simulation montrant l'erreur d'estimation de l'état. Une erreur ε est introduite à $t = 0s$ et $t = 100s$ et l'observateur est toujours capable de reconstruire l'état.

Le Projet du Contrôleur

L'analyse préliminaire pour l'élaboration de la loi de commande a été réalisée avec le modèle (7). En réalité, malgré le fait que le moteur diesel est un système fortement non linéaire, sa réponse en boucle ouverte est stable. Ceci dit, le premier pas consiste à trouver la commande u^* en régime permanent et l'état x^* correspondant. Puisque w et y^d sont connus, le régime stationnaire peut être obtenu à travers la solution du problème d'optimisation suivant:

$$u^*(w, y^d) := \arg \min_{u_d \in [u_{min}, u_{max}]} \|y_c(u_d, w) - y^d\|^2 \quad (11)$$

$$\begin{aligned} y_c(u_d, w) &= C(u_d, w)[\mathbb{I}_n - A(u_d, w)]^{-1} \cdot [B \cdot u_d + G \cdot w] + \varepsilon \\ x^*(u^*, w) &= [\mathbb{I}_n - A(u^*, w)]^{-1} \cdot [B \cdot u^* + G \cdot w] \end{aligned} \quad (12)$$

Une fois la commande u^* en régime permanent calculée, l'approche paramétrique consiste à trouver une simple paramétrisation temporel de la séquence de commandes futures. Tandis que dans les formulations classiques tous les composants de la commande sont utilisés comme variables de décision, l'approche paramétrique réduit considérablement le degré de liberté des variables inconnues tout en découpant du choix de l'horizon de prédiction:

$$\mathbf{u}(i\tau_s + t) = \text{Sat}_{u_{min}}^{u_{max}} \left(u^* + \alpha_1 \cdot e^{-\lambda \cdot i \cdot \tau_s} + \alpha_2 \cdot e^{-q \cdot \lambda \cdot i \cdot \tau_s} \right) \quad \text{for } t \in [(k-1)\tau_s, k\tau_s[\quad (13)$$

où $i \in \{0, \dots, N-1\}$, $\lambda > 0$, $q \in \mathbb{N}$ sont les paramètres de réglages, $\alpha_1, \alpha_2 \in \mathbb{R}^m$ sont les coefficients à déterminer. Les saturations sur les entrées sont structurellement respectées selon (4) avec l'opérateur *Sat*. En plus, deux autres contraintes peuvent être écrites comme suit:

$$u^* + \alpha_1 + \alpha_2 = u(k-1) \quad (14)$$

$$\alpha_1 \cdot (e^{-\lambda \cdot \tau_s} - 1) + \alpha_2 \cdot (e^{-q \cdot \lambda \cdot \tau_s} - 1) = p \cdot \delta_{max} \quad ; \quad p \in [-1, +1]^2 \quad (15)$$

où $p \in [-1, 1]^2$ est le vecteur de paramètres à déterminer. Il faut remarquer que l'expression (15) montre que la différence entre deux valeurs successives de commande ne doit pas dépasser la valeur maximale δ_{max} , selon la contrainte (5). Cela explique le pourquoi de $p \in [-1, 1]^2$. Nous pouvons donc obtenir un système avec 4 équations:

$$\begin{pmatrix} \alpha_1^{u_1}(p) \\ \alpha_2^{u_1}(p) \\ \alpha_1^{u_2}(p) \\ \alpha_2^{u_2}(p) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 & 0 & 0 \\ 0 & 0 & e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 \end{pmatrix}^{-1} \begin{pmatrix} u_1(k-1) - u_1^* \\ u_2(k-1) - u_2^* \\ p_1 \delta_{max}^1 \\ p_2 \delta_{max}^2 \end{pmatrix}$$

où $\alpha_1 = [\alpha_1^{u_1}; \alpha_1^{u_2}]$ et $\alpha_2 = [\alpha_2^{u_1}; \alpha_2^{u_2}]$ sont les 4 inconnues, solution du système linéaire ci-dessus. En injectant ces valeurs dans (13), nous obtenons finalement:

$$\mathbf{u}(i\tau_s + t, p) = Sat_{u_{min}}^{u_{max}} \left(u^* + \alpha_1(p).e^{-\lambda.i.\tau_s} + \alpha_2(p).e^{-q.\lambda.i.\tau_s} \right) \quad (16)$$

L'expression (16) montre clairement que le profil de commande dépend des paramètres p . Le prochain pas consiste à déterminer p à travers la solution d'un problème d'optimisation. Ceci passe par la définition d'une fonction coût qui pénalise l'erreur de poursuite $y - y^d$ et aussi l'état final. Nous pouvons donc définir le problème d'optimisation suivant:

$$\hat{p} := \arg \min_{p \in \mathbb{P}} \left[\rho_x \cdot \|X(N, \hat{x}(k), p) - x^*(y^d, w)\| + \sum_{i=0}^{N-1} \|Y(i, \hat{x}(k), p) - Y_f(i, y^d, w)\|_{Q_y}^2 \right] \quad (17)$$

où $\hat{x}(k)$ est l'état estimé à l'instant k , $Y(i, \hat{x}(k), p)$ et $X(i, \hat{x}(k), p)$ sont respectivement, la sortie et l'état à l'instant $k + i$ basé sur le profil de la commande en boucle ouverte défini par p dans $[k, k + N - 1]$ à partir de $\hat{x}(k)$. L'état $x^*(y^d, w)$ est l'état stationnaire calculé selon (12), $\rho_x > 0$ est le terme de pondération nécessaire pour renforcer la contrainte sur l'état final (pour des raisons de stabilité) et $Y_f(i, y^d, w)$ la séquence de consignes filtrées pour éviter les dépassements:

$$Y_f(i, y^d, w) = y^d + e^{-3\tau_s.i/t_r} \cdot [y(k) - y^d] \quad (18)$$

où t_r est le temps de réponse désiré du système en boucle fermée et Q_y la matrice de pondération définie comme étant:

$$Q_y = \begin{pmatrix} \frac{\rho_1}{\bar{y}_1} & 0 \\ 0 & \frac{\rho_2}{\bar{y}_2} \end{pmatrix}_{m \times m} \quad (19)$$

où ρ_1 et ρ_2 sont les termes de pondération sur y_1 et y_2 respectivement et \bar{y}_1 et \bar{y}_2 les termes de normalisation sur chaque sortie. La figure 4 illustre toute cette stratégie avec aussi la présence du MHO.

Pour l'instant, nous avons considéré que la solution stationnaire est donnée par la solution d'un problème d'optimisation à part (11). Toutefois, il est possible d'inclure la commande stationnaire en tant que variable de décision dans l'approche paramétrique. De cette façon, la nouvelle structure de la séquence de commande dévient

$$\mathbf{u}(i\tau_s + t, p) = Sat_{u_{min}}^{u_{max}} \left([p_3 \quad p_4]^T + \alpha_1(p).e^{-\lambda.i.\tau_s} + \alpha_2(p).e^{-q.\lambda.i.\tau_s} \right) \quad (20)$$

Il faut remarquer que cette fois-ci p est de dimension 4, à savoir, $p \in \mathbb{R}^4$. Dans ce cas, l'ensemble de contraintes à respecter sont $(p_1 \quad p_2)^T \in [-1, +1]^2$ et $(p_3 \quad p_4)^T \in \mathbb{R}^2$. Toutefois, afin de réduire l'espace de recherche pour p_3 et p_4 , des contraintes bornées peuvent être utilisées.

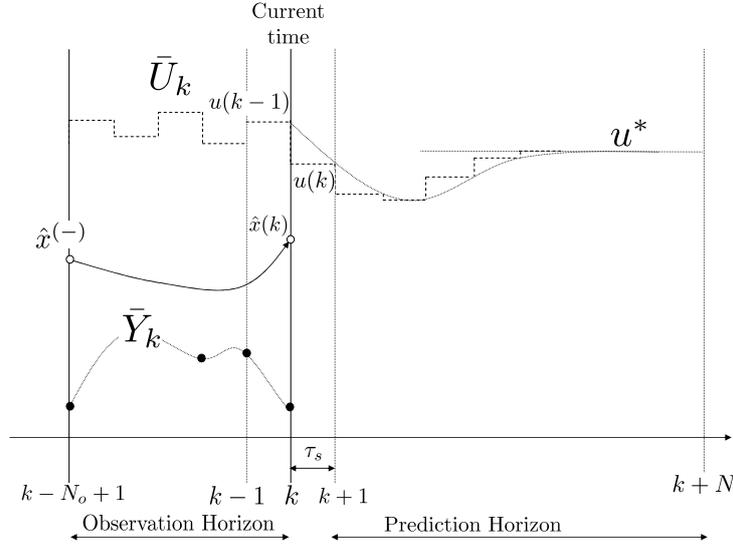


Figure 4: Notions concernant l'approche paramétrique et l'observateur à horizon glissant. L'instant actuel est k . \bar{Y}_k est collecté durant l'intervalle $[(k - N_o + 1)\tau_s, k\tau_s]$ et les valeurs passées de u et w sont stockées dans \bar{U}_k . Puis, l'état $\hat{x}(k)$ est obtenu à travers (10), qui est ensuite utilisée pour définir la paramétrisation exponentielle.

Une option possible consiste à considérer $(p_3 \ p_4)^T \in [u_{min}, u_{max}]^2$. Ceci revient à dire que les paramètres p_3 and p_4 doivent respecter les contraintes sur les entrées selon (4), et dans ce cas, ils remplacent la commande stationnaire u^* . Le système modifié pour la solution des paramètres α_1 and α_2 devient:

$$\begin{pmatrix} \alpha_1^{u_1}(p) \\ \alpha_2^{u_1}(p) \\ \alpha_1^{u_2}(p) \\ \alpha_2^{u_2}(p) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 & 0 & 0 \\ 0 & 0 & e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 \end{pmatrix}^{-1} \begin{pmatrix} u_1(k-1) - p_3 \\ u_2(k-1) - p_4 \\ p_1 \delta_{max}^1 \\ p_2 \delta_{max}^2 \end{pmatrix}$$

Toutefois, cette formulation nécessite quelques considérations concernant la fonction coût définie en (17). En réalité, celle ci dépend du calcul de x^* qui ne peut être obtenu qu'avec le modèle (7) puisqu'il y a une routine spécifique pour calculer l'état stationnaire (12). Cela n'est pas le cas pour le modèle non linéaire (6). Il faut donc changer la pondération finale sur l'état en (17), qui dans ce cas peut être définie comme la norme de la différence entre deux états successifs à la fin de l'horizon de prédiction, afin d'adresser la stabilité. Une autre possibilité consiste tout simplement à faire $\rho_x = 0$, puisque le système est stable en boucle ouverte. Ce choix semble être très raisonnable selon les résultats de simulations obtenus. Par conséquent, le calcul de p ne dépend plus d'une structure particulière du modèle utilisé, et ceci fait que l'approche paramétrique puisse être appliquée comme une solution du type *black-box* pour les moteurs diesel.

Le dernier point à considérer consiste à choisir une procédure d'optimisation pour le calcul de p . Nous proposons une méthode basée sur la Programmation Quadratique Séquentiel (SQP), gradient et région de confiance. Pour des raisons de simplicité, nous allons la décrire très brièvement. La première routine, appelée SQP, est une mise-à-jour de la $i^{\text{ème}}$ composante de p à travers une approximation quadratique scalaire. Le succès ou non de cette recherche augmente ou réduit la région de confiance et conduit à une variation de p_i . En plus, ces variations permettent d'avoir aussi une expression approchée du gradient. Pour cela, une deuxième routine appelée *Gradient* est utilisée pour permettre une descente du gradient sur l'ensemble des composantes de p selon une deuxième région de confiance. Lorsque le nombre d'évaluation du critère dépasse la valeur maximale n_{fe} , les itérations s'arrêtent et la valeur courante de p est passée dans $p(t_{k+1})$. Malgré sa simplicité, cette méthode s'est montrée assez puissante en obtenant le même niveau de performance par rapport à d'autres routines classiques du type Powell ou Simplex.

Résultats

Cette section présente quelques résultats obtenus en simulation ainsi que la validation expérimentale de l'approche paramétrique sur le banc d'essais de l'Université de Linz ¹. Le scénario dans les deux cas est le cycle *New European Driving Cycle* (NEDC), qui est censé représenter l'usage typique des voitures en Europe et est aussi utilisé pour l'analyse des niveaux des émissions des moteurs en général. En plus, tous les programmes relatifs à la CPNL et MHO ont été développés en langage C et intégrés à l'environnement MATLAB à travers les S-fonctions.

La figure 5 montre les simulations en utilisant le modèle non linéaire (6). La configuration hardware utilisée est un PC Pentium 4, 3GHz. Le système a été échantillonné à 10 ms, avec $N = 100$. Les coefficients λ et q des exponentielles en (16) valent 8 et 1.25 respectivement et le nombre maximum d'évaluation de fonction $n_{fe} = 4$. Notez que le NEDC exige une poursuite très rapide des valeurs de MAF et MAP et le contrôleur arrive bien à suivre les consignes désirées. Il faut aussi noter que, lorsque les set-points ne sont pas faisables, les vannes sont saturées dans leur position maximale. Dans ce cas, il faut trouver un bon compromis entre les variables ou sinon privilégier une variable par rapport à l'autre en agissant sur les paramètres ρ_1 et ρ_2 .

Chronologiquement, le modèle (7) a été livré par les autrichiens avant le modèle (6) et pour cette raison, le premier a été utilisé pour la validation expérimentale. Les résultats sont montrés dans la figure 6. Ce modèle a été identifié à 50 ms et l'horizon de prédiction utilisé cette fois-ci était $N = 30$. Puisqu'il s'agit du modèle (7), l'observateur est nécessaire, et l'horizon d'observation N_o utilisé étant égal à 10. En plus, la pondération sur l'état final doit être aussi prise en compte, avec $\rho_x = 1e3$. Notez que les contraintes sur la position des vannes sont respectées et

¹Moteur diesel BMW M47TUE

la poursuite des set-points de MAF et MAP est assez correcte. Les incertitudes du modèle et l'échantillonnage à 50 ms peuvent expliquer le dépassement à $t \approx 200s$. Pour cette raison, la prochaine étape consiste à valider cette approche en utilisant le système (6), vu qu'il s'agit d'un modèle plus complet et représentatif.

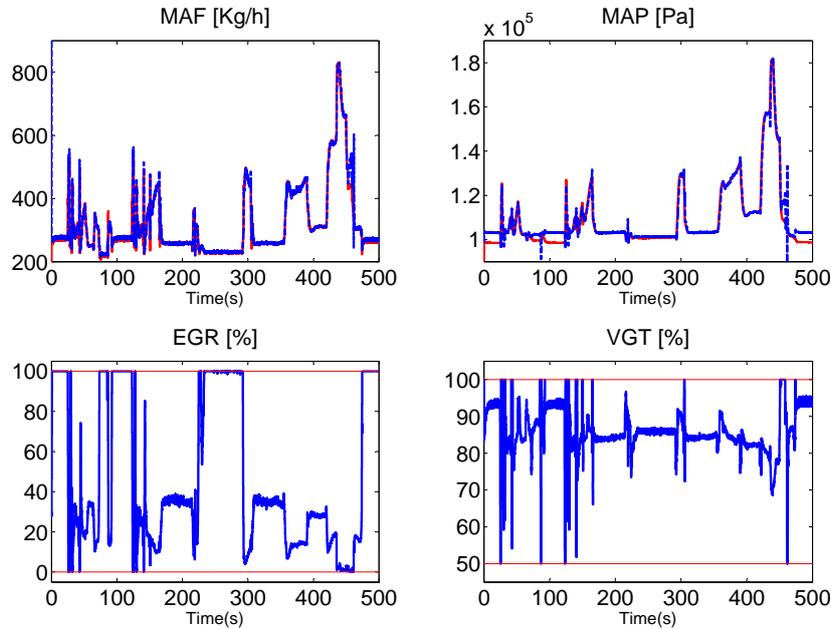


Figure 5: Résultats de simulation de l'approche paramétrique avec le NEDC. Modèle (6).

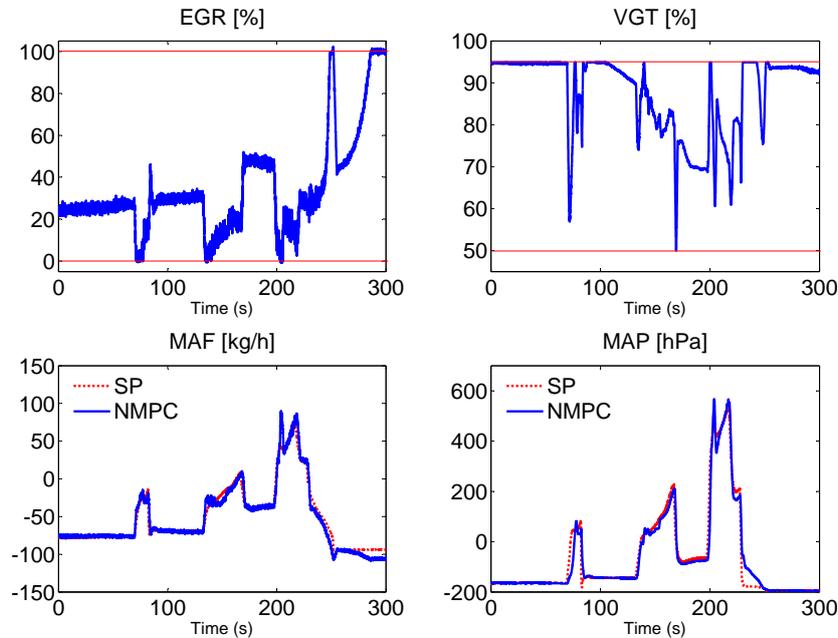


Figure 6: Résultats expérimentaux de l'approche paramétrique relative à la partie haute-vitesse du NEDC. Modèle (7).

La CPNL Paramétrique pour les Pendules Jumeaux

Le Système des Pendules Jumeaux

Le deuxième exemple d'application concernant l'approche paramétrique est le système des pendules jumeaux sur un chariot. Ce système est composé de deux pendules avec des inerties différentes, attachés à un chariot. L'objectif de la commande consiste à amener les deux pendules à la position verticale d'équilibre, à travers le couple moteur qui agit sur le chariot. La méthode a été validée expérimentalement sur la plateforme des pendules du département d'automatique de l'Université de Grenoble.

Le schéma de la plateforme utilisée dans les expériences est montré dans la figure 7. Les notations m_i , l_i , I_i , θ_i sont, respectivement, les masses, les longueurs par rapport au centre, moments d'inertie et position angulaire du i_{eme} pendule, r la position du chariot par rapport à l'origine O_o et r_{max} la valeur maximale admissible pour r . La force F appliquée sur le chariot est générée par la commande u , qui représente le couple livré par le moteur.

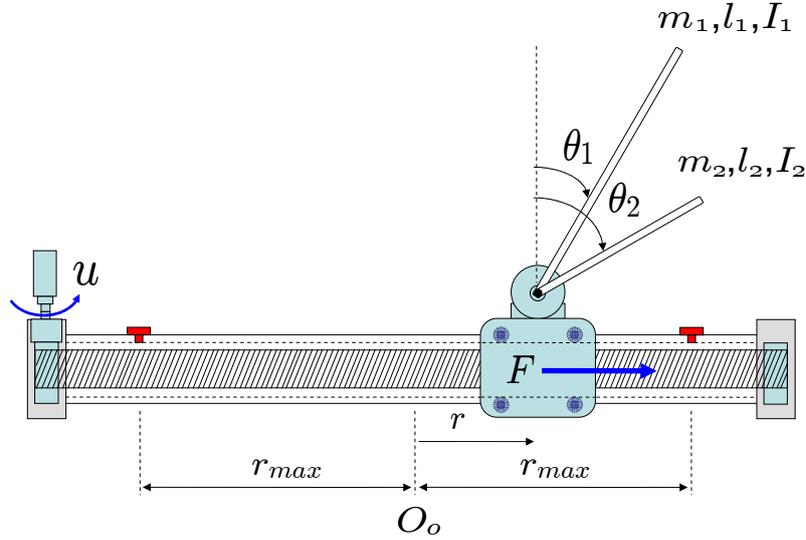


Figure 7: Le système des pendules jumeaux (Département d'automatique, Université de Grenoble).

Le modèle du système est obtenu à travers le développement des équations de Lagrange, ce qui donne les équations suivantes:

$$\begin{cases} \ddot{\theta}_i = -\alpha_i v \cos \theta_i + \beta_i \sin \theta_i & ; \quad \alpha_i = \frac{m_i l_i}{m_i l_i^2 + I_i} & ; \quad \beta_i = g \alpha_i & ; \quad i \in \{1, 2\} \\ \ddot{r} = v \end{cases} \quad (21)$$

$$u = \alpha_0^m(\theta_1, \theta_2)v + \beta_0^m(\dot{r}, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \quad (22)$$

où α_0^m et β_0^m sont les valeurs des inerties équivalentes dépendant des paramètres physiques du système pendule-moteur, v l'accélération du chariot et u la variable de commande qui représente le couple moteur. L'objectif de la loi de commande consiste à stabiliser l'ensemble suivant:

$$\{x \in \mathbb{R}^6 | x = (r^f \ 0 \ 0 \ 0 \ 0 \ 0)^T; r^f \in [-r_{max}, +r_{max}]\} \quad (23)$$

tout en respectant les contraintes sur la commande et sur la position du chariot:

$$u(t) \in [u_{min}, u_{max}] \quad ; \quad r(t) \in [-r_{max}, +r_{max}] \quad (24)$$

Le Projet du Contrôleur

L'élaboration de l'approche paramétrique passe par la définition du concept de niveau d'énergie E d'un pendule et sa dérivée \dot{E} . Ces expressions peuvent être représentées par les relations suivantes:

$$E_i = \frac{1}{2}\dot{\theta}_i^2 + \beta_i(\cos\theta_i - 1) \quad ; \quad \dot{E}_i = -\alpha_i v \dot{\theta}_i \cos\theta_i \quad ; \quad i \in \{1, 2\} \quad (25)$$

Nous pouvons donc souligner quelques caractéristiques intéressantes concernant le niveau d'énergie. Premièrement, pendant les instants où $v = 0$, la valeur de \dot{E}_i vaut 0, c-à-d, les deux pendules possèdent un niveau d'énergie constant. En plus, si à un instant donné nous obtenons $E_i = 0$, alors les futures trajectoires sous $v = 0$ mènent à l'état $\theta_i = \dot{\theta}_i = 0$ ce qui représente la position verticale d'équilibre. Deuxièmement, dans les cas où $\dot{\theta}_i \cos\theta_i \neq 0$, E est sensible à v et donc, en agissant sur v , il est possible d'agir sur la dérivée de E afin de diminuer le niveau d'énergie de façon incrémentale. En revanche, si $\dot{\theta}_i \cos\theta_i \approx 0$, E_i devient insensible à v . En tout cas, il est très rare que $(\dot{\theta}_1 \cdot \cos\theta_1)$ et $(\dot{\theta}_2 \cdot \cos\theta_2)$ s'annulent au même moment dû aux différentes caractéristiques physiques des pendules et par conséquent, il est donc possible d'améliorer le niveau d'énergie d'un pendule si l'autre reste insensible à v . L'approche paramétrique va donc se baser sur la méthode d'énergie, déjà invoquée dans [Åstrom and Furuta, 2000]. Pour ce faire, les définitions suivantes sont proposées:

$$E_i^n = \frac{|E_i|}{E_i^{max}} \quad ; \quad i \in \{1, 2\} \quad ; \quad E = \max\{E_1^n, E_2^n\} \quad (26)$$

où E_i^n représente le niveau d'énergie normalisé et E sa valeur maximale qui correspond au niveau d'énergie total du système. Avant de formaliser la loi de commande, il faut d'abord adresser le problème des contraintes. En ce qui concerne la contrainte sur la position, un simple changement de variables est proposé selon:

$$v(k) = -K_s \begin{pmatrix} r(k) - r_d(k) \\ \dot{r}(k) \end{pmatrix} \quad (27)$$

où $K_s \in \mathbb{R}^{1 \times 2}$ représente le gain stabilisant pour l'asservissement de la position désirée r_d , qui dévient la nouvelle variable de commande. En effet, r_d doit être choisie tel que $r_d \in [-r_{max}, +r_{max}]$. En plus, K_s est calculé pour que les transitoires respectent la région admissible $[-r_{max}, r_{max}]$ et aussi de sorte que les niveaux d'accélération, et par conséquent le couple moteur, ne dépassent jamais les saturations $[u_{min}, u_{max}]$. Ceci règle donc les contraintes sur l'actionneur. Nous pouvons définir un profil constant par morceaux pour r_d comme suit:

$$r_d(k\tau_s + t) = r_d(k) \quad ; \quad t \in [0, T_p] \quad (28)$$

où T_p est l'horizon de prédiction $T_p = N\tau_s$. Désormais, nous allons considérer que T_p est choisi plus grand ou égal au temps de réponse t_r du chariot, donné par le choix de K_s dans l'expression (27). Plus précisément, cette condition assure que à la fin de l'horizon de prédiction $v \approx 0$, ce qui garantit que, lorsque l'horizon de prédiction est décalé, la valeur finale de E reste constante (en absence de frottement) si nous gardons la position actuelle. Ceci dit, nous pouvons définir une fonction coût qui fait le lien à l'état du système à la fin de l'horizon de prédiction:

$$J(x(k), p) = E(F(x(k), T_p, p)) \quad (29)$$

où $F(x(k), T_p, p)$ est la solution de (21) à l'instant T_p , et p le paramètre à optimiser défini comme étant $p = r_d$. L'étape suivante consiste à trouver une stratégie d'optimisation pour le paramètre p . Les routines classiques d'optimisation peuvent être très lourdes pour l'exécution dans le temps d'échantillonnage. Dans ce cas, l'idée est de vérifier trois possibilités: rester dans la position actuelle, se déplacer à gauche ou se déplacer à droite. Ce déplacement, défini comme d_r , est calculé en fonction du niveau d'énergie du système selon une fonction linéaire du type $d_r(E) = d_{r_0} + \delta_r E$. Cela veut dire que pour $E = 1$, la valeur maximale pour d_r est appliquée et diminue au fur et mesure que E diminue jusqu'à la valeur minimale permise d_{r_0} . Par conséquent, l'ensemble de valeurs admissible pour la variable de décision p est donné par:

$$\mathbb{P} := \left\{ \max\{-r_{max}^{RHC}, r - d_r\}, r, \min\{+r_{max}^{RHC}, r + d_r\} \right\} \quad (30)$$

où r_{max}^{RHC} est l'arrêt virtuel défini tel que $r_{max}^{RHC} < r_{max}$, utilisé pour éviter que le transitoire du contrôleur linéaire (qui sera expliqué dans la suite) ne viole pas les contraintes de position. Pour cette raison, r_{max}^{RHC} ne concerne que la partie relative à l'horizon fuyant, à savoir, *Receding Horizon Control* (RHC). Avec cette définition, nous pouvons établir le problème d'optimisation à résoudre à chaque période d'échantillonnage:

$$\hat{p}(x(k)) = \arg \min_{p \in \mathbb{P}} J(x(k), p) \quad (31)$$

où \hat{p} représente la valeur optimale de p . La figure 8 montre une vision intuitive du choix de p et la contraction de E . En injectant \hat{p} dans l'expression (27), nous obtenons la loi de commande prédictive suivante:

$$v(k) = K_{RHC}(x(k)) = -K_s \begin{pmatrix} r(k) - \hat{p}(x(k)) \\ \dot{r}(k) \end{pmatrix} \quad (32)$$

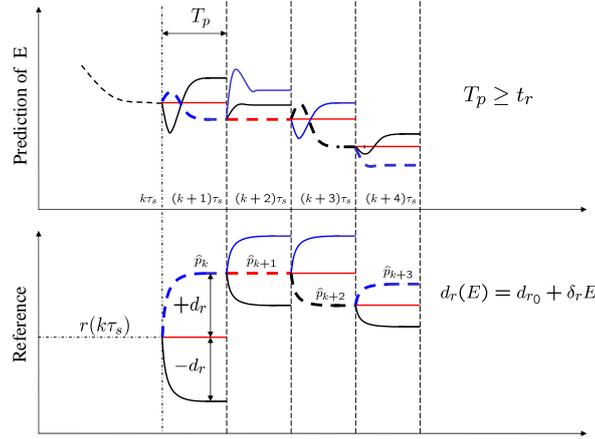


Figure 8: Schéma intuitif de la contraction de E selon les valeurs optimales \hat{p} .

La prochaine étape consiste à définir la loi de commande au voisinage de la position d'équilibre local et les conditions nécessaires pour basculer d'un contrôleur à l'autre. D'abord, écrivons le système linéarisé du système autour de $x = 0$:

$$\begin{cases} \ddot{\theta}_1 = -\alpha_1 v + \beta_1 \theta_1 \\ \ddot{\theta}_2 = -\alpha_2 v + \beta_2 \theta_2 \\ \ddot{r} = v \end{cases} \quad (33)$$

ce qui décrit un système linéaire du type $\dot{z} = Az + Bv$, où z est la nouvelle variable d'état $z = (\dot{r} \ \theta_1^m \ \dot{\theta}_1 \ \theta_2^m \ \dot{\theta}_2) \in \mathbb{R}^5$, et $\theta_1^m, \theta_2^m \in [-\pi, +\pi]$ sont égaux à θ_1, θ_2 , modulo 2π . Il faut remarquer que la position finale r^f n'est pas imposée. Pour ce système, un régulateur linéaire classique peut être utilisé pour obtenir un gain K_L , et par conséquent, une loi de commande discrète du type $v(k) = -K_L \cdot z(k)$, qui stabilise les pendules à la position verticale haute d'équilibre. Cependant, cette loi ne peut être utilisée qu'autour d'un petit voisinage \mathcal{A}_0 de l'état désiré. Pour bien définir cette région, il est normal d'utiliser une fonction de Lyapunov telle que $\mathcal{A}_0 = \{z \in \mathbb{R}^5 | z^T S z \leq \rho_0\}$, où $\rho_0 = z_0^T S z_0$ et z_0 représente l'état qui définit les frontières autour de la position verticale où le modèle linéaire semble être plus acceptable. Donc, tant que le système reste en dehors de \mathcal{A}_0 le basculement vers le contrôleur linéaire ne peut pas être réalisé. En plus cette condition n'est pas suffisante, puisque le transitoire le $v(k)$ peut violer les contraintes de position r_{max} . Pour essayer de contourner ce problème, une deuxième condition doit être additionnée afin de surveiller toutes les

positions intermédiaires du chariot, sous la présence de $v(k)$. En réalité, la prédiction de la valeur maximale de ces positions doit être évaluée, à savoir $\varphi(x(k))$. Donc, $\varphi(x(k)) < r_{max}$ est la deuxième condition à respecter pour pouvoir changer la loi de commande, ce qui nous donne un contrôleur du type hybride. Nous pouvons donc résumer cette stratégie de la façon suivante:

$$v(k) = \begin{cases} -K_L z(k) & \text{si } z^T(k)S z(k) \leq \rho_0 \text{ et } \varphi(x(k)) < r_{max} \\ K_{RHC}(x(k)) & \text{sinon} \end{cases}$$

Un observateur a été aussi développé pour le système de pendules jumeaux, afin d'obtenir un estimateur d'état au cas où les mesures de vitesses (chariot et pendules) ne sont pas disponibles et si les mesures des capteurs d'angles sont biaisées. Un observateur non linéaire a été proposé pour élaborer une stratégie du type retour de sortie, où les biais sont correctement estimés, et l'erreur d'estimation converge vers zéro. Un résultat très important a montré que le système peut être très sensible lorsque les biais sur les capteurs d'angles varient légèrement à faible amplitude et fréquence. Le chariot, dans ce cas, devient très oscillant pour garder le système en équilibre, ce qui explique le comportement dans la validation expérimentale, vu qu'il existe beaucoup d'incertitudes dans le modèle nominal. Les détails concernant cet observateur sont disponibles dans [Murilo and Alamir, 2007].

En ce qui concerne la validation expérimentale, quelques considérations doivent être faites afin d'adapter la solution théorique à l'implémentation temps-réel. En réalité, l'effet du frottement est très présent sur le vrai système. Pour cela, une identification en boucle ouverte est proposée à fin d'obtenir les coefficients de frottement sec et visqueux qui agissent sur les articulations des pendules. Deuxièmement, le choix de l'horizon de prédiction doit être le plus petit possible, c'est à dire, $T_p = t_r$ afin de minimiser l'effet des incertitudes et le frottement sur le système. Finalement, plus d'itérations peuvent être réalisées, au lieu de trois, pour essayer de profiter au maximum le temps de calcul disponible pour intégrer le système. En fait, le nombre d'évaluations de fonction n_{fe} multiplié par le coût d'intégration du système τ_{comp} doit être inférieur au temps d'échantillonnage τ_s , c'est à dire, $n_{fe} \times \tau_{comp} \leq \tau_s$.

Résultats

Les résultats obtenus en simulation sont montrés dans la figure 9, avec $T_p = 0.6s$, $\delta_r^{max} = 0.4m$, $r_{max}^{RHC} = 0.6m$, $r_{max} = 0.7m$, $\tau_s = 0.05s$, $d_{r_0} = 0.1m$, $\alpha_1 = 0.67m^{-1}$ et $\alpha_2 = 1.33m^{-1}$. Notez bien la décroissance de E et le profil d'accélération v du chariot. Pour un temps de réponse t_r donné, la contrainte sur la commande est respectée en agissant sur δ_r^{max} , ou vice-versa. La figure 10 montre les résultats expérimentaux obtenus dans la plateforme des pendules jumeaux du département d'automatique de l'Université de Grenoble, cette fois-ci avec $\alpha_1 = 2.59m^{-1}$, $\alpha_2 =$

$3.51m^{-1}$ et avec un temps d'échantillonnage plus petit $\tau_s = 0.01s$. L'expérience démarre à 5 s. Notez bien que l'approche paramétrique arrive globalement à faire descendre le niveau d'énergie du système. Notez aussi le comportement oscillant du chariot lorsque les pendules sont en position verticale. Ceci est principalement dû au frottement et aux incertitudes du modèle. Comme mentionné précédemment, ce genre de comportement a déjà été noté lors de simulations réalisées en présence d'un faible biais variable introduit sur les capteurs d'angles.

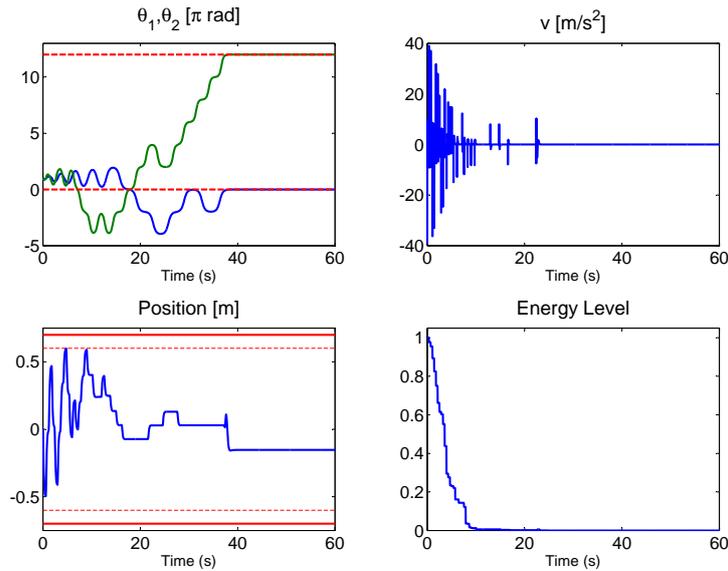


Figure 9: Évolution du système en boucle fermée avec $\delta_r^{max} = 0.4m$.

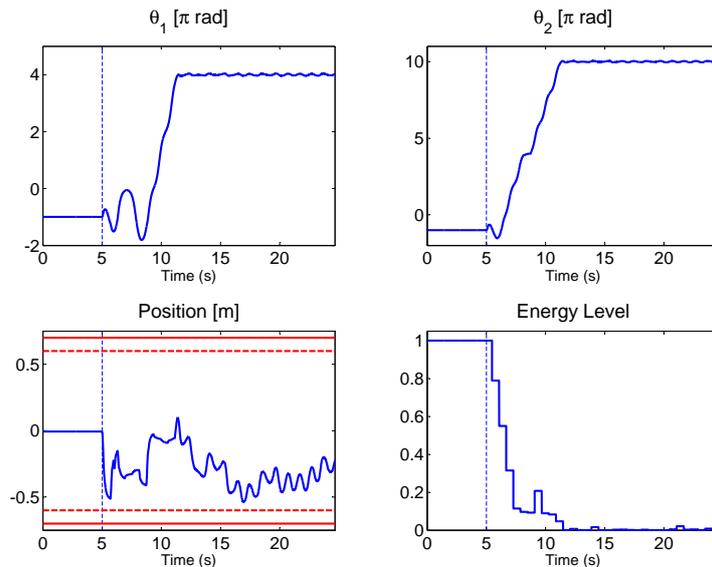


Figure 10: Résultats expérimentaux expriment l'efficacité de l'approche paramétrique pour stabiliser de système des pendules jumeaux.

Conclusion

Dans cette thèse, deux schémas à faible dimension basés sur l'approche paramétrique ont été proposés pour deux systèmes non linéaires à dynamiques rapides: le moteur diesel et les pendules jumeaux. Ces solutions ont été validées expérimentalement et ont montré l'efficacité de la méthode. De plus, dans le cas du moteur diesel, l'approche paramétrique peut être utilisée comme une solution du type *boite noire*, totalement indépendant de la complexité du modèle non linéaire. Ceci peut être généralisé pour les systèmes qui possèdent des caractéristiques similaires au comportement du moteur diesel. En ce qui concerne la résolution des problèmes d'optimisation, un algorithme basé sur SQP/Gradient/Région de confiance a été développé et utilisé pour l'application du moteur diesel. Et finalement, deux observateurs non linéaires ont été proposés pour les deux systèmes, avec l'implémentation temps-réel d'un observateur à horizon glissant pour le moteur diesel.

Chapter 1

Introduction

1.1 General Aspects of Model Predictive Control

Model Predictive Control (MPC) is an advanced control methodology which uses explicitly the system model to predict the future evolution of the process. The basic idea of MPC consists in computing an optimal control sequence over a prediction horizon at each decision instant, by minimizing some given cost function expressing the control objective. The first control signal is scheduled to be applied to the system during the next sampling period and this optimization process is successively repeated at each sampling time. The prediction horizon keeps being shifted forward and for this reason MPC is also called receding horizon control (RHC). Figure 1.1 illustrates a simplified view of MPC.

In the last recent years, the interest of the scientific community in MPC has considerably grown as well as the number of successful industrial case studies involving MPC controllers. Originally conceived to be applied to slow dynamic processes such as chemical industries or oil refineries, MPC-like strategies are becoming more and more present in different industrial sectors. This was made possible thanks to the advances in computer science and to some extent, the dedicated formulations that aims at using MPC for systems with fast dynamics. In 2006, the workshop *MPC for Fast Nonlinear Systems* prior to the *45th IEEE Conference on Decision and Control* [Findeisen, 2006] showed some interesting data concerning the publications and applications involving MPC:

- 2120 publications in 2004-2005 (Inspec).
 - Adchem 2006, 30 % of submitted control papers are related to MPC.
-

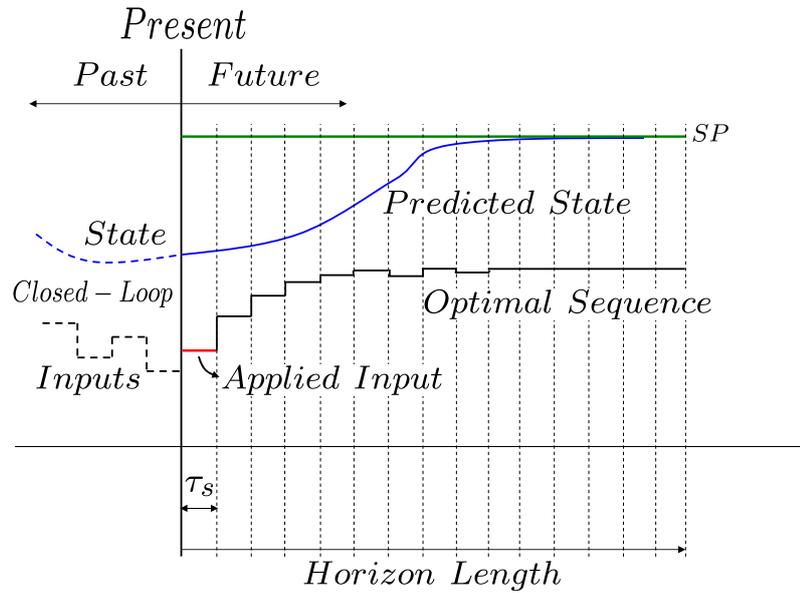


Figure 1.1: Schematic view of the MPC strategy. An optimization process returns the optimal sequence computed in open loop over some prediction horizon. Only the first element of the resulting control profile is applied during the next sampling period τ_s . Repeating this process at each sampling instant results in a state feedback.

- Several special issues in journals under preparation.
- More than 5800 known implementations spanning a wide range of application areas: petrochemical industry, food industry, aerospace industry, car industry, etc.

An excellent survey containing several aspects of MPC is presented in [Mayne et al., 2000], which highlights its different properties and limitations. Here, the main points are presented by discussing the advantages and drawbacks of this control design methodology.

Handling Constraints

The first advantage of MPC is the fact that constraints on the inputs and the states can be explicitly handled in the problem formulation. These constraints such as control saturation, safety requirement and contractual specifications (to cite just a few examples) are unavoidable in any realistic context. The presence of these constraints in control problems renders quite often MPC the only available option for the success of the control task.

Dealing with Nonlinearities

Although linear MPC theory has reached a high level of maturation, there are several situations where nonlinearities are quite strong and real processes can not be correctly represented by linear models. Moreover, even for linearized systems, improvements on the required performance due to a more restrictive and demanding specifications in industry, may considerably change the admissible operational region where linear assumptions are valid. All these reasons make linear paradigm insufficient to describe the whole complexity of the process, and nonlinear approaches become more appropriate. This motivates the use of a more elaborated strategy, namely nonlinear model predictive control (NMPC).

The main difference between MPC and NMPC approaches lies in the use of nonlinear models, which can be empirical (identified input-output or state space based models), physical (also called first principle models) or gray box (intermediate level between empirical and physical). In recent years, the number of theories based on NMPC formulations has considerably grown. Obviously, the problem complexity is much greater comparing to the linear case. Thus, one of the main difficulties of the NMPC strategy lies in the fact that the convexity of the optimization problem is no more guaranteed as in the linear case. Nonlinear Programming (NLP) approaches become a natural candidate to deal with such problems. However, NLP-based solutions may be very difficult to obtain. This is an important issue for NMPC and more details about it are widely discussed in the following chapters.

Real-time Implementation

Another important issue in NMPC is the computation time needed to perform the optimization process that leads to the optimal control sequence. This is a crucial point for predictive control, especially when fast systems are considered, since the optimal solution must be calculated within a short sampling period when systems with fast dynamic are addressed.

The first practical applications of MPC were mostly developed for processes where the dynamics were slow enough to perform *all* the required computations. This explains why slow-dynamic processes, such as the one encountered in the petrochemical industries, were the first real application where MPC could be effectively applied. Since then, several theoretical issues have been developed in order to design faster and feasible NMPC methods to be applied with higher sampling rates.

Robustness

As in any control design, the issue of robustness against model mismatches and/or disturbances has to be considered. Until now, there is no satisfactory answer to

this problem in the MPC context despite the huge efforts of a part of the scientific community [Magni and R.Scattolini, 2007]. It is even suggested in [Alamir, 2006b] that the fact that MPC approaches heavily depends on long terms prediction makes them particularly vulnerable to model mismatches. Coupling MPC to local analytical controllers in a inner loop may probably be the good option.

1.2 Motivations

This PhD work addresses the challenging problem of applying NMPC to nonlinear dynamical systems showing fast dynamics. By "fast dynamics", it is meant that the sampling period that is used to update the control input is sufficiently short to prevent the complete solution of the underlying optimization problem when the latter is designed using standard schemes.

More specifically, this PhD work promotes the *parameterized approach in NMPC design* [Alamir, 2006a] as a candidate solution to this challenging issue. In particular, the validation of the proposed ideas on concrete experimental facilities is a main streamline of this work since such a validation is the ultimate step.

In the following section, the outline of the present dissertation is given.

1.3 Outline of the Dissertation

The dissertation is organized in five chapters.

Chapter 2 proposes a general literature review on NMPC. First, a historical background of predictive control is presented. Then, some preliminary notions on NMPC are introduced and important points such as stability and real-time implementation are emphasized.

Chapter 3 focuses on the state of the art of NMPC strategies for fast systems. The main issues, advantages and drawbacks of the main approaches are discussed in details. Moreover, some important aspects of the parameterized NMPC scheme are presented and compared with the standard NMPC methods.

Chapter 4 presents the parameterized NMPC scheme for a diesel engine air path. It is shown that the resulting control scheme not only meets the real-time requirements but also handles the constraints and realizes nice tracking performances as

well. Moreover, the proposed control strategy uses the model as a black-box simulator since no particular structure related feature is used in the development of the algorithm. This makes the algorithm a candidate solution to be used with more advanced and precise models of the process. Results from experimental validation of a real-world diesel engine test bench are shown to emphasize the efficiency of the parameterized NMPC scheme. The strategy is completed by designing a moving horizon observer (MHO) in order to estimate the unmeasured states.

Chapter 5 proposes a second real application of the parameterized NMPC: the swing-up and stabilization of the twin-pendulum system. This system consists of two decoupled pendulums with different inertias fixed on a cart. The control problem is to stabilize both pendulums in the upward vertical at rest position by acting on the cart acceleration. A parameterized NMPC scheme is proposed to deal with nonlinearities, constraints on both the actuators, the state (cart excursion) and on-line computation requirements.

Finally, a general conclusion of this dissertation is presented and further issues are discussed for future work.

Chapter 2

Nonlinear Model Predictive Control

2.1 Introduction

In this chapter, an overview of the main topics concerning the NMPC is presented together with some mathematical background. During the recent years, the development of new NMPC strategies witnessed a considerable progress in several domains. Here, two of them are more emphasized: the stability problem and the real-time implementation. For the first one, the sufficient conditions to guarantee asymptotic stability for nonlinear systems are presented in a condensed form. The second one introduces some important notions about the NMPC real-time implementation, which is the main part of this dissertation. Especially for fast dynamic systems, the real-time aspects of NMPC represent a crucial point to take into account in the problem formulation.

This chapter is organized as follows. First, in section 2.2, a historical background is proposed in order to track the evolution of the optimal and predictive control strategies through time. Then, in section 2.3, some preliminary notions about NMPC are introduced together with the mathematical background in order to go deeper into the NMPC theory. In section 2.4, the sufficient conditions to guarantee stability for general NMPC schemes are presented. Section 2.5 addresses the real-time issue and the stability conditions in such a context are formalized. Finally, section 2.6 gives the conclusion of this chapter.

2.2 Historical Background

The first steps of predictive control go back to the late 1950's where the Bellman's principle of optimality was stated [Bellman, 1957]. It is quite important to underline the link between predictive and optimal control since the basic principle of predictive control is to solve an optimal control problem at each sampling period. Belleman introduced an important property for optimal strategies which says that the remaining part of an optimal trajectory remains optimal for the optimal control problem defined on the remaining part of the prediction horizon. Afterwards, some relevant observations made by Kalman in [Kalman, 1960] for linear systems showed that *optimality does not imply stability* and important concepts such as Control Lyapunov Functions (CLF) have been introduced to analyze stability of closed-loop systems. Some years later, in 1967, the first guidelines of predictive control were roughly announced in [Lee and Markus, 1967], which suggested that only the first portion of an optimal sequence must be applied and a new scenario must be re-computed at each decision instant. In fact, this clearly represents the main idea of predictive control.

The second important step for the development of predictive control was made in the seventies, with the advances in the computer science and the development of microprocessors. The latter has considerably increased the computation speed as well as the memory storage and provided several improvements on software programming and the possibility to embed mathematical tools. All these advances resulted in the development of important industrial applications of control design, especially for predictive control which is highly dependent on the computational resources to solve optimization problems on line. Actually, industrial processes represent a very rich field of predictive control applications not only because of the several constraints present in real processes, but also due to the fact that industry constantly needs to optimize performances.

Therefore, at the end of the seventies, the first MPC applications were developed for the industry, most of them for chemical processes which were slow enough to ensure feasible computations of optimal solutions. Basically two kinds of algorithms were proposed. The algorithm IDCOM (Identification and Command), presented in [Richalet et al., 1978, 1976], and the DMC (Dynamic Matrix Control) [Cutler and Ramaker, 1980], both based on heuristic methods, also known by Model Predictive Heuristic Control (MPHC). This first generation of algorithms had a great impact in the industry but they were not able to manage the constraints yet. Such limitations lead to the development of an important second class of algorithms, headed by the Quadratic Dynamic Matrix Control (QDMC), for linear systems with quadratic cost functions [Garcia and Morshedi, 1986]. The main feature of such algorithms was the introduction of the important concept of Quadratic Programming (QP) to solve

constrained problems for linear systems under quadratic cost functions in open loop. Since then, several kinds of MPC algorithms were developed for the industry. For more details about these approaches, an excellent survey of a wide range of industrial applications of MPC can be found in [Qin and Badgwell, 2003].

Initially, linear MPC approaches were used as the simplest way to formulate the control problem. This is because the identification process of linear models is quite simple. Moreover, the computation of the quadratic cost function can be easily obtained, providing convex problems for which the solutions were well studied and solved using many commercially available products. However, many reasons make the linear approaches inappropriate to meet the increasing performance requirements in the industry. Correct handling of strong nonlinearities and non-convex optimization problems have become the only issue to correctly address the real problems and NMPC quickly arose to fill this important gap in the industry. As a result, in recent years some important softwares such as Interior Point Optimizer (IPOPT) and Multiple Shooting Code for Optimization (MUSCOD) were developed in order to address predictive control for constrained nonlinear systems [Wächter and Biegler, 2006, Leineweber, 1996]

Despite the success of NMPC applications in the industrial context, the theoretical aspects of stability were not explicitly taken into account in such formulations. The pragmatic point of view of engineers as well as the open-loop stable behavior of most processes (which is not a sufficient condition for closed-loop stability) may explain why the stability problem was not explicitly addressed for a while. Then, in 1988, the contribution of Keerthi and Gilbert proved the asymptotic stability for NMPC by means of Lyapunov methods for discrete-time systems and under mild technical conditions [Keerthi and Gilbert, 1988]. Two years later, a similar proof was provided for the continuous-time case [Mayne and Michalska, 1990]. This was definitely the starting point for the development of several important contributions on NMPC. Next chapter is specifically dedicated to discuss some of these contributions. Before that, it is quite necessary to introduce some mathematical background and important definitions related to NMPC. This is the aim of the next section.

2.3 Problem Formulation

2.3.1 Some Definitions and Notation

In this section, the mathematical background is presented in order to formalize some basic concepts related to NMPC. Let us consider a general state space representation of a plant described by a discrete-time nonlinear system as follows:

$$x(k+1) = f(x(k), u(k)) \quad (2.1)$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ the control input, both defined at the present instant k , $x(k+1)$ the state at instant $k+1$ and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$. For the sake of simplicity, the system (2.1) can also be represented as $x_{k+1} = f(x_k, u_k)$ or even by the short notation $x^+ = f(x, u)$. This representation can be easily obtained from the differential equations of the model, by converting them into a set of first-order equations.

In this presentation, we assume that the control objective is to stabilize the origin $x = 0$ and we assume that the pair $x = 0, u = 0$ is a stationary pair for the dynamic given by (2.1). In addition, the control input u is constrained according to:

$$u(k) \in \mathbb{U} \subset \mathbb{R}^m \quad (2.2)$$

where $\mathbb{U} \subset \mathbb{R}^m$ represents a compact and convex set of admissible values of $u(k)$. Considering that the control input is piece-wise constant. A general control sequence \mathbf{u} composed of N successive input values can be defined as follows:

$$\mathbf{u} := (u(k) \quad u(k+1) \quad \cdots \quad u(k+N-1)) \in \mathbb{U}^N \subset \mathbb{R}^{N \cdot m}$$

The above representation can also be written in a condensed form:

$$\mathbf{u} = \mathcal{U}_{pwc}(\cdot, \{u(i)\}_{i=k}^{k+N-1})$$

where \mathcal{U}_{pwc} represents the classical piece-wise constant (PWC) control profile for \mathbf{u} , under the sampling period τ_s . More formally, this control sequence can be defined as follows:

$$\mathbf{u}(i\tau_s + t) = u(i) \text{ for } t \in [0, \tau_s[\quad , \quad i \in \{k, \dots, k+N-1\}$$

Moreover, the admissible subset \mathbb{U} is supposed to contain the origin, namely $0 \in \text{int}(\mathbb{U})$, which means that the set \mathbb{U} contains the asymptotically desired control value, namely $u = 0$. The control objective is to find a stabilizing control law which respects the input constraints defined in (2.2) and steers the state trajectory to 0, while respecting the state constraints defined by:

$$x(k) \in \mathbb{X} \subset \mathbb{R}^n \quad (2.3)$$

where \mathbb{X} is a closed and convex set. A final constraint on the state may be added in order to force a weighting term on $x(k+N)$, leading to:

$$x(k+N) \in X_f \subset \mathbf{X} \quad (2.4)$$

where X_f is a compact subset of \mathbf{X} .

2.3.2 The Cost Function

In NMPC context, one aims at stabilizing the system at the origin using trajectories that minimize some cost function that reflects the control objective. In order to manipulate such trajectories, let us denote by $x^{\mathbf{u}}(k)$ the future state trajectory when some control profile \mathbf{u} is applied to the system (2.1), starting from current state $x(k)$:

$$x^{\mathbf{u}}(k) := (x(k) \quad x(k+1) \quad \cdots \quad x(k+N)) \in \mathbb{R}^{N \cdot n}$$

Since the dynamic system is supposed to be time invariant, the index k may be omitted in the further developments and the time base is set to 0. For this reason, the state value at $t = (k+i)\tau_s$ under the control sequence \mathbf{u} is represented by the short notation $x(i)$ instead of $x(k+i)$.

A general form of a cost function with finite horizon and terminal constraint can be expressed as follows:

$$V_N(x, \mathbf{u}) = F(x(N)) + \sum_{i=0}^{N-1} L(x(i), u(i)) \quad (2.5)$$

where $F(x(N))$ is the terminal cost and $L(x(i), u(i))$ a weighting term called stage cost, which loosely expresses the desired closed-loop performance. In this case, N is also called the prediction horizon. The weighting term L is generally chosen such that the following technical assumption is satisfied:

$$L(x, u) \geq c \cdot \left\| \begin{pmatrix} x \\ u \end{pmatrix} \right\|^2 \quad \text{with } c > 0 \quad (2.6)$$

Actually, the above condition, defined by the euclidian norm, is not that difficult to fulfill since a simple quadratic term can be added to any original formulation to meet (2.6).

2.3.3 The NMPC State Feedback

The NMPC strategy consists in computing at each sampling period the best future control sequence by minimizing the cost function defined in (2.5) under the constraints given by (2.2)-(2.3)-(2.4).

More precisely, at each decision instant k , the solution of the following optimization problem $\mathcal{P}_N(x(k))$ (that depends on the state x) has to be obtained:

$$\begin{aligned} \mathcal{P}_N(x) \quad : \quad \hat{\mathbf{u}}(x) &:= \arg \min_{\mathbf{u}} [V(x, \mathbf{u})] \\ \text{subject to} \quad : \quad \mathbf{u} &\in \mathbb{U}^N, \quad x(i) \in \mathbb{X}, \quad x(N) \in X_f \end{aligned} \quad (2.7)$$

where $x(i)$ denotes the state of the system i sampling instants later when the piecewise control sequence \mathbf{u} is applied to the system.

The optimal solution $\hat{\mathbf{u}}(x)$ is the sequence of future control actions that minimizes the cost function while meeting the problem's constraints. Note that since the optimization problem itself depends on the current state x , the optimal solution is also a function of the current state.

In what follows, the following notation is used to refer to the successive actions in the optimal sequence:

$$\hat{\mathbf{u}}(x) = (\hat{u}^{(0)}(x), \dots, \hat{u}^{(N-1)}(x)) \quad , \quad \hat{u}^{(i)}(x) \in \mathbb{R}^m$$

According to the definition of the predictive control strategy, the first control in the sequence $\hat{\mathbf{u}}(x)$, namely $\hat{u}^{(0)}(x)$, is applied to the system during the time interval $[k, k+1[$ resulting in the following discrete-time state feedback:

$$\kappa_N(x) = \hat{u}^{(0)}(x) \quad (2.8)$$

Moreover, the optimum value $V(x, \hat{\mathbf{u}}(x))$ of the cost function (2.5) can be represented by the short notation:

$$\hat{V}_N(x) = V(x, \hat{\mathbf{u}}(x)) \quad (2.9)$$

This quantity plays a crucial role in the proof of the stability of the closed-loop system as it is shown in the next section.

2.4 Stability Analysis

In this section, the sufficient conditions that guarantee asymptotic stability of the closed-loop nonlinear system given by:

$$x^+ = f(x, \kappa_N(x))$$

are shown. This presentation is largely inspired by [Mayne et al., 2000] from which only the main streamlines are presented since this stability analysis is not the main point in the present PhD work.

Let us consider the future state trajectory under the optimal control sequence $\hat{\mathbf{u}}$, namely $\hat{\mathbf{x}}$, at the present instant k .

$$\hat{\mathbf{x}} = \{\hat{x}, \hat{x}^{(1)}, \dots, \hat{x}^{(N)}\}$$

According to the definition of the NMPC control law, the control $\kappa_N(x) = \hat{u}^{(0)}(x)$ is applied during the sampling period $[k, k+1]$. At instant, $k+1$, the state $x^+ = \hat{x}^{(1)}$ is reached and the problem to be solved now is precisely $\mathcal{P}_N(x^+)$ which is defined over the prediction horizon $[k+1, k+N+1]$.

The idea leading to the sufficient conditions for stability is to take as a candidate solution for $\mathcal{P}_N(x^+)$ the remaining sequence of controls

$$\{\hat{u}^{(1)}(x), \dots, \hat{u}^{(N-1)}(x)\} \quad (2.10)$$

completed by a final action such that the resulting sequence is admissible for the constrained optimization problem $\mathcal{P}_N(x^+)$. This is made possible by the following assumption:

Assumption 2.1. *For all $\xi \in X_f$, there exists a control $\kappa_f(\xi) \in \mathbb{U}$ such that $f(\xi, \kappa_f(\xi)) \in X_f$. In other words, there exists a local control law κ_f defined in X_f taking values in \mathbb{U} with respect to which the terminal set X_f is positively invariant.*

Under assumption 2.1, the terminal action to be added at the end of the shifted sequence (2.10) is precisely given by the value of the feedback κ_f at the final state x_N , namely:

$$\tilde{\mathbf{u}} = \{\hat{u}^{(1)}(x), \dots, \hat{u}^{(N-1)}(x), \kappa_f(\hat{x}^{(N)})\}$$

which leads to the following state trajectory $\tilde{\mathbf{x}}$ starting at instant $k+1$:

$$\tilde{\mathbf{x}} = \{\hat{x}^{(1)}, \dots, \hat{x}^{(N)}, f(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)}))\}$$

The sequence $\tilde{\mathbf{u}}$ is clearly admissible since each one of its actions is clearly in the admissible control set \mathbb{U} , the final state is clearly in the admissible set $\mathbb{X} \cap X_f$ and the preceding states belong to the admissible set by construction (optimal trajectory of the preceding problem $\mathcal{P}_N(x)$). Therefore, since $\tilde{\mathbf{u}}$ is an admissible control sequence, it comes by the very definition of optimality that any optimal solution, say $\hat{\mathbf{u}}(x^+)$ to the optimization problem $\mathcal{P}_N(x^+)$ satisfies:

$$V_N(\mathbf{x}^+, \hat{\mathbf{u}}(x^+)) \leq V_N(\mathbf{x}^+, \tilde{\mathbf{u}}) \quad (2.11)$$

Using the short notations

$$\hat{V}_N^+ = V_N(\mathbf{x}^+, \hat{\mathbf{u}}^+) \quad ; \quad \tilde{V}_N^+ = V_N(\mathbf{x}^+, \tilde{\mathbf{u}})$$

It comes that definition of $\tilde{\mathbf{u}}$:

$$\tilde{V}_N^+ = F(f(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)}))) + \sum_{i=1}^{N-1} L(x(i), u(i)) + L(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)}))$$

The sum of the stage cost L in the above expression can also be defined as:

$$\sum_{i=1}^{N-1} L(x(i), u(i)) = \sum_{i=0}^{N-1} L(x(i), u(i)) - L(x, \kappa_N(x))$$

which leads to the following relation:

$$\tilde{V}_N^+ = \hat{V}_N - L(x, \kappa_N(x)) - F(\hat{x}^{(N)}) + L(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)})) + F(f(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)}))) \quad (2.12)$$

and using the notation,

$$\Delta F(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)})) = F(f(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)}))) - F(\hat{x}^{(N)})$$

the expression (2.12) becomes:

$$\tilde{V}_N^+ = \hat{V}_N - L(x, \kappa_N(x)) + \Delta F(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)})) + L(\hat{x}^{(N)}, \kappa_f(\hat{x}^{(N)}))$$

and finally using the notation $\xi = \hat{x}^{(N)} \in X_f$, the following expression is obtained:

$$\tilde{V}_N^+ \leq \hat{V}_N^+ = \hat{V}_N - L(x, \kappa_N(x)) + [\Delta F + L](\xi, \kappa_f(\xi)) \quad (2.13)$$

It becomes then clear that in order to be able to prove the decrease of the positive function \hat{V}_N over the closed-loop trajectory, the following assumption is appropriate:

Assumption 2.2. *The terminal cost F , the stage cost L , the final subset X_f and the local control law κ_f defined by the assumption 2.1, satisfy the following inequality:*

$$[\Delta F + L](\xi, \kappa_f(\xi)) \leq 0$$

for all $\xi \in X_f$.

Indeed, if the assumption 2.2 holds, the inequality (2.13) becomes:

$$\hat{V}_N^+ \leq \hat{V}_N - L(x, \kappa_N(x))$$

or by re-using the original notations:

$$\hat{V}_N(x^+) \leq \hat{V}_N(x) - L(x, \kappa_N(x)) \quad (2.14)$$

which clearly proves the asymptotic stability of the origin provided that the stage cost $L(x, u)$ is positive definite with respect to x .

It is worth noting that assumption 2.2 simply states that the final weighting term F is a Control Lyapunov Function (CLF) under the local control κ_f since one clearly has:

$$F(f(\xi, \kappa_f(\xi))) - F(\xi) \leq -L(\xi, \kappa_f(\xi)) \leq 0$$

The following proposition summarizes the sufficient conditions for the asymptotic stability of the closed-loop system:

Proposition 2.1. *If the following conditions are satisfied:*

- *The functions f , F and L are continuous*
- *L satisfies (2.6)*
- *\mathbb{U} is compact (needed for the existence of optimal solution)*
- *\mathbb{X} and $X_f \subset \mathbb{X}$ are closed*
- *X_f is positively invariant under κ_f*
- *F is a CLF under κ_f over X_f*
- *The assumptions 2.1 and 2.2 are satisfied*

Then the predictive controller is well defined and the corresponding closed-loop system is asymptotically stable. More precisely, the Lyapunov function is precisely the optimal cost \hat{V}_N .

The sufficient conditions invoked in proposition 2.1 are those suggested in the seminal paper [Mayne et al., 2000] where a particularly interesting unified view has been proposed. Before this unification, there were a quite impressive amount of

particular works that introduce, each, different set of sufficient conditions. This resulted in a traditional classification of the different approaches according to the kind of stability-oriented constraints as well as the specific treatment of the prediction horizon.

The only NMPC formulations that do not fit the above framework are those where the stability arguments are based on the property of contraction [Morari and Oliveira, 2000, Alami, 2007]. In these formulations, the prediction horizon is a decision variable and is therefore computed together with the optimal control sequence when solving the underlying optimization problem. The stability proof is therefore slightly different from the streamline depicted above. The reader can refer to the above cited references for more details.

2.5 Real-Time Issue

2.5.1 Distributing the Optimization over the System Life-Time

In the preceding sections, the stability of the closed-loop system has been considered assuming that the optimization problem can be almost *instantaneously* solved. By instantaneously, it is meant that the time needed to solve the optimization problem can be considered as negligible when compared to the characteristic time of the system being considered.

In fact, as mentioned in the introduction, this assumption is particularly true for chemical processes, whose slow dynamics cope perfectly with the real time requirements to perform optimization process. For this reason, NMPC was successfully implemented in that context. Dedicated softwares were able to solve demanding optimizations problems, with thousands of decision variables and constraints.

Unfortunately, this scenario is no more valid when dealing with dynamical systems showing fast dynamics. In this case, the sampling period is drastically reduced and the available computation time may not be sufficient to perform the optimization task in within the sampling period.

For this reason, real-time optimization has become one of the most challenging problems concerning NMPC. In this section, some basic concepts are introduced in order to state the underlying problem in a rather informal way. The existing solutions to cope with this implementation problem are discussed in the next chapter, where an extensive literature review on the different NMPC approaches related to

this issue are provided.

When dealing with nonlinear programming, there is no closed analytical form giving the optimal sequence that solves the optimization problem. Instead, an iterative process has to be applied that takes the following form:

$$\hat{\mathbf{u}}^{(i)} = \mathcal{S}(\hat{\mathbf{u}}^{(i-1)}, x) \quad (2.15)$$

where \mathcal{S} is some optimizer that differs according to the solver being used and i is the iteration index. The control sequence must be initialized with some initial value $\mathbf{u}^{(0)}$. Theoretically, the optimal solution is given by:

$$\hat{\mathbf{u}}(x) = \lim_{i \rightarrow \infty} \hat{\mathbf{u}}^{(i)} \quad (2.16)$$

However, for real time implementation, the iterations must stop after $iter_{max}$ iterations

$$\hat{\mathbf{u}}(x) = \hat{\mathbf{u}}^{(iter_{max})} \quad (2.17)$$

It is worth noting that the number of iterations that would be needed to achieve the optimization task depends not only on the current state x but also on the desired precision. Moreover, the time needed to perform $iter_{max}$ iterations must be lower than the available sampling time τ_s .

In order to deal with this problem, the idea of distributing the optimization over the system life-time has been first suggested by [Alamir, 2001] and applied to the minimum interception time problem. More precisely, it has been shown that by performing only one iteration of a descent algorithm, an NMPC law can be derived that leads to a remarkably robust behavior under high uncertainties on the aerodynamic coefficients.

Since this early proposition, many researchers adopted this simple and intuitive idea [Ohtsuka, 2004, Alamir, 2006a, Diehl et al., 2005a, DeHaan and Guay, 2007] to cite only few works. These works are discussed in the next chapter. In the remainder of this chapter, an intuitive and rather informal set of sufficient conditions for such a solution to lead to an asymptotically stable closed-loop behavior are discussed.

2.5.2 The Extended System

Note that under the real-time constraint, it is no more possible to practically deal with the implicitly defined quantity $\hat{\mathbf{u}}(x)$ used in section 2.4 since it is no more

possible to compute this optimal sequence. Instead, one has to deal with the current estimation of the optimal sequence that evolves in time according to (2.15). Assuming that i_t iterations of the solver can be performed at each sampling period, one can write the following extended system:

$$\begin{cases} \mathbf{u}^+ = \mathcal{S}^{i_t}(\mathbf{u}, x) \\ x^+ = f(x, u^{(0)}) \end{cases} \quad (2.18)$$

in which the state of the system and the sub-optimal control sequence evolves simultaneously. Moreover, the control applied to the system is exactly the first control in the sequence $\hat{\mathbf{u}}$ which is no more linked to the state by an implicit static relation.

One is clearly in the presence of an extended state:

$$\begin{pmatrix} x \\ \mathbf{u} \end{pmatrix}$$

where the control sequence \mathbf{u} becomes an internal state. The dynamic of this state is driven by the iteration of the solver \mathcal{S} that still depends on the current state x .

It goes without saying that the analysis of the stability of the extended system (2.18) is much more involved than the stability analysis in the ideal case. This is because in the classical proof of stability, the optimality of the sequence $\hat{\mathbf{u}}(x)$ played a quite crucial role. In the real-time iteration case, it is very difficult to assess the quality of an intermediate result $\mathbf{u}^{(i)}$ obtained after a finite number of iterations and starting from some initial value that may be far from the optimal solution. That is the reason why the current section suggests some intuitive arguments to highlight the intuitive conditions under which the whole thing may work.

Proposition 2.2 (Informal).

If the following conditions are satisfied:

1. *The procedure \mathcal{S} is sufficiently efficient,*
2. *The sampling time is sufficiently small,*
3. *The original predictive control formulation is stabilizing in the ideal context*
4. *The system shows no finite escape time.*

Then, the state of the extended system (2.18) converges to a small neighborhood of the origin.

An informal proof of this rather intuitive statement is proposed in [Alamir, 2006b]. Roughly speaking: if the first two assumptions are satisfied, the decrease in the optimal cost, between two sampling instants is *stronger* than the potential increase that would result from the change in the state that parameterizes the optimization problem. This result in a net decrease of the optimal cost. When the optimal cost reaches a neighborhood of the true minimum, the arguments used in the proof of stability in the ideal case prevail and the third assumption may be used to conclude. However, for this whole process to take its time, it is mandatory that the system state remains bounded during the transient. That is why the fourth assumption is used.

A formal proof including roughly the same arguments has been used in [Diehl et al., 2005b] in the particular case where the multiple-shooting based real-time iteration scheme (see chapter 3) proposed by [Diehl et al., 2005a] is used. The proof is more technical but the basic arguments are the same.

2.6 Conclusion

This chapter provided a brief survey of some important points related to NMPC. The stability issue of a NMPC scheme was analyzed in details, and conditions for stability were also formalized following the guidelines presented in [Mayne et al., 2000]. Moreover, the real-time aspects were also invoked in the problem formulation and an informal stability proof of the resulting system was also proposed. This proof falls in the quite recent idea of distributing the optimization problem over the system life-time in order to deal with the limitation of the available computation time to proceed optimization process, which is a critical point when fast dynamic systems are considered. The next chapter analyzes the main contributions that share this common idea.

Chapter 3

NMPC for Fast Systems

3.1 Introduction

In this chapter, the different NMPC formulations which were developed to handle fast dynamic systems are discussed. Roughly speaking, fast systems are those showing response time smaller than the time needed to solve the underlying open-loop optimization problem in a standard way. Actually, this situation arises in several real applications, and it is particularly frequent in robotics and mechatronic systems.

This chapter is divided in two parts. The first (section 3.2) provides the state of the art on the main strategies for the implementation of NMPC schemes for fast systems, by discussing the advantages and drawbacks of each strategy. The second (section 3.3), focuses specifically on the low-dimensional parameterized NMPC approach, since it is the strategy adopted in this work. Finally in section 3.4, a general discussion is proposed.

3.2 Fast NMPC: State of the Art

There are several ways to classify the different NMPC formulations that intend to tackle the real-time implementation issue. Here, a first classification is obtained by splitting the existing works into two main families: the linear and nonlinear formulations. A brief introduction of each category is proposed, and the most representative contributions of each one are discussed in the sequel. However, the nonlinear approaches are more detailed since they represent the target of this dissertation.

3.2.1 Linear Formulations

Several linear MPC strategies fall in this category such as the explicit off-line feedback computation approach based on the Piece-Wise Affine (PWA) approximation [Bemporad et al., 2002] and the recently developed on-line active set approach for PWA models [Ferreau et al., 2006]. Roughly speaking, as far as nonlinear systems are concerned, these approaches address the real-time requirement by replacing the original problem by a new and generally different one which possesses a highly structured form that lends itself to efficient computations.

Although these linearization-like approaches attempt to solve a modified problem that can be quite different from the original one (because of the use of PWA models that can be quite different from the physical nonlinear model), they encountered and still have a huge popularity. For this reason, these two important contributions in this domain are detailed in the following.

3.2.1.1 Explicit approaches

The explicit approach, also called off-line method, has become a successful MPC design tool for constrained linear systems. A complete overview of explicit approaches is well detailed in [Alessio and Bemporad, 2008]. One of the main contributions in this domain was presented in [Bemporad et al., 2002], which also inspired some others advances in the explicit MPC [Borrelli et al., 2003, Tøndel et al., 2003]. Many applications were developed as well as a hybrid toolbox software for predictive control [Ortner and del Re, 2007, Bemporad, 2004].

The key idea of the proposed strategy is to perform off-line computations in order to solve a standard parameterized QP-problem. Thus, the resulting solution can be explicitly calculated taking into account the constraints on the states and inputs. It is a fact that having this off-line preparation reduces considerably the amount of on-line computation but also shows some drawbacks which are discussed later.

The main point of [Bemporad et al., 2002] is the fact that the state space \mathbb{X} can be subdivided into some polyhedral critical regions (CR), on each of which the optimal active set is constant, and that the explicit solution of the parametric QP on each of these critical regions is an affine function of x_0 [Zafiriou, 1990].

Figure 3.1 illustrates a simplified view of a two-dimensional state space partitioning. The constraints represented by the inequalities $C1 \leq 0, \dots, C5 \leq 0$ are affine functions of the state x and they are used to define the critical region CR_0 and the multiple convex subsets R_i . In fact, CR_0 is a polyhedron in the x -space, and

represents the largest set of $x \in \mathbb{X}$ such that the combination of active constraints at the minimizer remains unchanged. Once the critical region CR_0 has been defined, the rest of the space $CR^{rest} = \mathbb{X} - CR_0$ has to be explored and new critical regions generated.

The algorithm runs off-line to cover the region of interest of the state space and the optimal solution of a multi-parametric (mp) QP is calculated for each subset. Once the background calculations are finished, the on-line task consists in finding the current subset according to the current state. This reduces substantially the computation time, since there is no more on-line optimization to be done. This is the major advantage of the explicit approach as only the critical regions corresponding to the current value of state must be found. The corresponding pre-computed optimal gain is then used to compute the control to be applied.

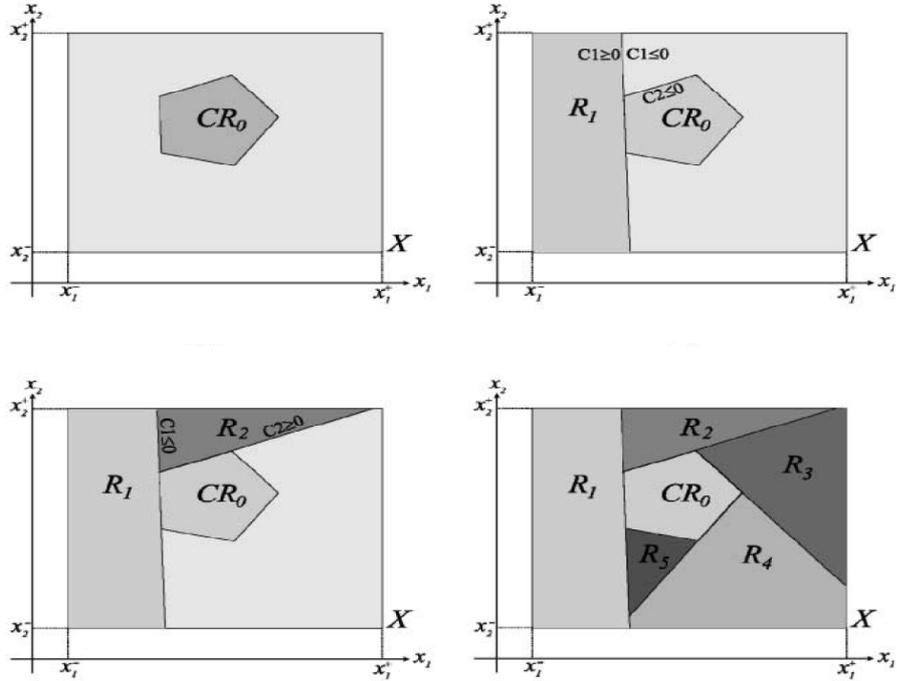


Figure 3.1: Schematic view of the partitioning of the state space in the explicit approach [Bemporad et al., 2002]. The subsets and the critical regions are calculated off-line. The controller must only compute the right subset the current state belongs to in order to apply the pre-computed optimal solution.

However, the exponential growth of the off-line computation requirements in the parameter size, as well as the storage of the partition description, limit considerably the dimension of the system that can be handled and the number of uncertainties or additional parameters to take into consideration. Moreover, the number of regions may be dissuasive and can even make the on-line determination of the current region

a quite involved task in some cases. In [Pannocchia et al., 2006], a method to reduce the off-line complexity of the explicit approach is proposed provided suboptimal online performance are accepted.

3.2.1.2 Online active set strategy

The potential drawbacks of the explicit solution inspired the development of different approaches. An online active set strategy was proposed in [Ferreau et al., 2006]. The main idea of this approach lies in the fact that the active set does not change very much from one QP to the next one.

In this method, a straight line in the state space links the past value of the state to the new one. Then, the corresponding QPs on this line can be solved since such set remains convex. The QP solution depends affinely on the initial state x_0 as long as the critical region does not change, but it might happen that the boundaries of the critical regions are crossed. The basic idea of the algorithm, which has been previously proposed by [Best, 1996], considers the homotopy for parametric QP, as can be seen in the first scenario of figure 3.2, which shows a simplified view of the homotopy paths. The figure illustrates the trajectory of the state, driven by some continuation parameter $\tau \in [0, 1]$, through the different critical regions.

One of the main advantages of the proposed method is the fact that one produces a sequence of optimal solutions for QPs on the homotopy path. This is a very important feature to ensure real-time feasibility, since it is possible to stop such sequence after every partial step and re-start a new homotopy path from the current value towards the next QP. This is illustrated in the second scenario of figure 3.2. While in the first case there is no limitation related to the changes of the active sets, for the second one, a maximum active set changes may be imposed per QP, where only two active set changes are permitted, in order to guarantee real-time implementation.

This on-line active set algorithm motivated the development of the software *qpOASES* [Ferreau, 2007], which was successfully implemented in some practical applications [Ferreau et al., 2007]. Indeed, this methodology not only avoids the huge off-line preparation but also addresses the real-time problem. However, the present solution keeps the idea that the whole control input vector is taken as a degree of freedom, which may represent a very strong requirement, especially when fast systems are considered. Moreover, the active set method, as well as the explicit approach presented previously, are limited to only linearized model or originally linear systems. This motivates the study of more intrinsically nonlinear approaches.

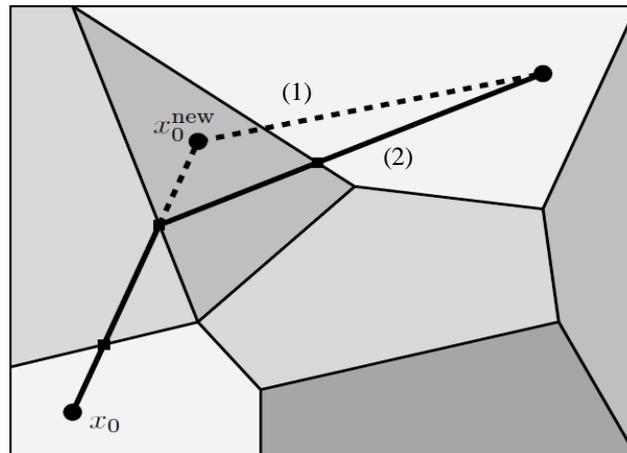


Figure 3.2: Schematic view of two scenarios concerning the homotopy paths [Ferreau et al., 2006]. The algorithm must solve the QP(x_0^{new}) from the initial solution QP(x_0), by moving from x_0 to x_0^{new} according to a normalized continuation parameter τ . This may keep the feasibility of the resulting solution for the intermediate points. The second scenario shows that the number of active set changes was limited in order to address real-time requirements.

3.2.2 Nonlinear Formulations

The success of the linearization-based controllers for predictive control was principally due to the fact that the complete solution of the original nonlinear constrained problem would be anyway unfeasible within the available computation time. This difficulty, for a long time considered as insuperable, inspired the idea of distributing the optimization process over the system lifetime [Alamir, 2001]. Later, several approaches emerged in this direction [Ohtsuka, 2004, Diehl et al., 2005a, Alamir, 2006a, DeHaan and Guay, 2007] that are discussed here.

First, a brief overview of the off-line standard dynamic optimization is presented. According to [Cervantes and Biegler, 2009], there are two types of NLP approaches: the indirect and the direct methods.

- **Indirect Methods**

Also known as variational approach, these methods are based on the first order necessary condition for optimality, obtained from the Pontryagin's Maximum Principle [Pontryagin et al., 1962]. Such conditions can be written as a two-point boundary value problem for problems without inequality constraints and classical solvers can be used. However, several drawbacks, such as the difficulty to handle inequality constraints and the often unstable dynamic of the adjoint

state differential equation make indirect methods less appropriate for on-line NMPC.

- **Direct Methods**

These methods are based on piecewise constant controls and are widely used for NMPC. They can be separated into two main groups: sequential (or control parametrization) and simultaneous strategy

- In the sequential approach, the decision variable is the sequence of control actions to be applied. The cost function is obtained by integrating the system equation using the current guess of the control sequence. The problem formulation is easy to construct using reliable differential algebraic equations (DAE) solvers as well as NLP solvers. However, repeated numerical integration of DAE may considerably increase the computation time especially when inappropriate values of the control sequence are encountered during the optimization. Indeed, these sequence may lead to stiff differential systems that may increase the computation time or, even worse, lead to unstable behavior.
- In the simultaneous approach, both the control input sequence and the state trajectory are considered as decision variables to be found simultaneously. This is the approach adopted in the multiple-shooting algorithm [Bock and Plitt, 1984] and the collocation method [Biegler, 1984]. The system equations are transformed into constraints linking the decision variables and this results in a generally extremely high dimensional NLP problem that is sparse and dedicated solvers have to be used. The stability issue of the direct method is overcome at the price of a larger problem and a more involved problem formulation. However, the over sophisticated software that are involved in the solution may not be compatible with on-board facilities and even memory limitation.

The real-time concerned formulations that have been proposed for efficient implementation of NMPC can be divided into five families:

- Real-Time Iteration with Multiple Shooting
 - Continuation/Generalized Minimum Residual (C/GMRES)
 - Interior Point-based Methods
 - Hybrid-Parameterized Scheme
 - Parameterized NMPC approach
-

These approaches deserve a special attention since they cover the main existing streamline of current research on NMPC schemes dedicated to fast systems. More details on these strategies are presented in the following section. The low-dimensional parameterized NMPC scheme is more discussed in section 3.3, since it is the approach adopted in this PhD work.

3.2.2.1 Real-Time Iteration with Multiple Shooting

In [Diehl et al., 2005a], a real time iteration strategy has been developed in order to distribute the optimization process over the system life-time. The computation of the optimal control sequence is based on the multiple shooting method, a simultaneous approach which offers several advantages in the real-time context.

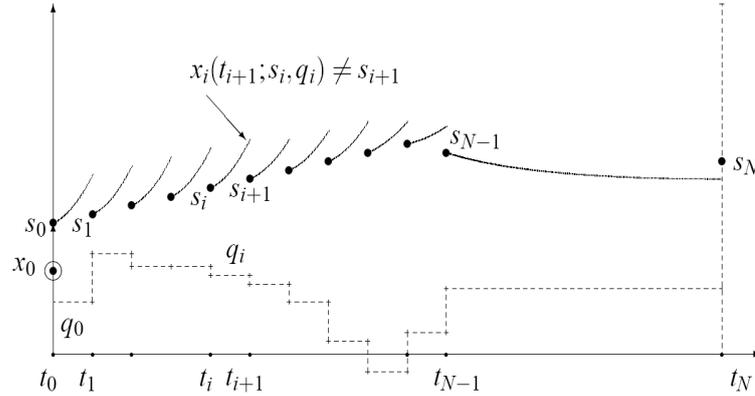


Figure 3.3: Schematic view of the multiple shooting strategy applied to NLP problems [Bock and Plitt, 1984, Diehl et al., 2005a]. Multiple shooting is an efficient way to initialize optimization problems, to deal with highly nonlinear and unstable systems, and treat with path constraints.

Multiple shooting strategy, introduced in [Bock and Plitt, 1984], basically considers a piecewise constant representation of the control profile using a temporal grid $\{t_i\}_{i=0}^N$, along the prediction horizon. The ordinary differential equation (ODE) representing the system dynamics are solved over each interval $[t_i, t_{i+1}]$ numerically, starting from an artificial initial value of the state s_i , and computing the future trajectories $x_i = f(t_{i+1}, s_i, q_i)$, where q_i is the control input. The corresponding stage cost L is computed and then integrated for each interval. Figure 3.3 illustrates the multiple shooting strategy. By doing so, the intermediate states s_i are also part of the decision variable (degrees of freedom) to be computed in order to solve the optimal control problem.

One advantage of this multiple shooting approach lies in the stability of the integration scheme since transient unfeasible trajectories can be used in the NLP solver. More precisely, if a *bad* control sequence is encountered, the system is not integrated from the initial state until the end of the prediction horizon, only pieces of trajectories are computed starting from the currently adopted values of the state s_i . Moreover, constraints can be explicitly imposed on the state over the time interval without being implicitly depending on the control sequence, at least temporarily.

Only one descent iteration is applied at each sampling period. The computation of the ingredients needed to compute the descent direction is done in two steps. During the current sampling period, the part of the problem that does not depend on the state is computed (preparation phase), then as soon as the new measurement arrives, the QP problem description is completed and a descent direction can be performed. The resulting control is applied and the process is repeated during the next sampling period. As a result, this approach consists in two important phases, the preparation and the feedback phase, which is shown in figure 3.4.

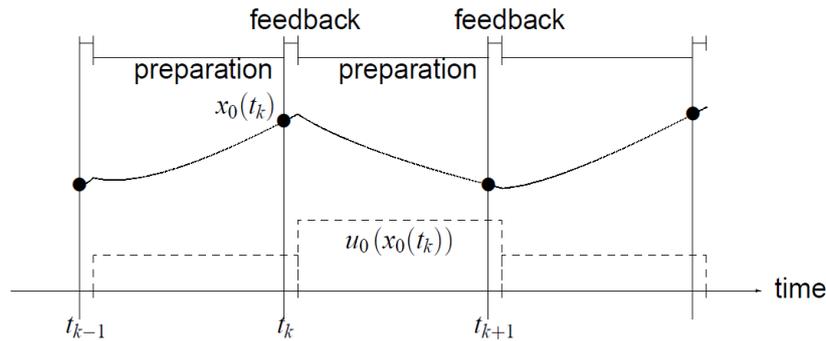


Figure 3.4: Schematic view of the preparation and the feedback phases present in the real-time iteration method [Diehl et al., 2005a]. Note the difference between the two phases concerning the computational effort.

The decision variable to be updated at each feedback phase is composed of the control sequence, the state sequence and the adjoint state sequence defined over the prediction horizon, namely

$$y = (\lambda_k, s_k, u_k, \lambda_{k+1}, s_{k+1}, u_{k+1}, \dots, \lambda_{k+N}, u_{k+N})$$

where $s_i \in \mathbb{R}^n$ is the state values at instant i , $\lambda_i \in \mathbb{R}^n$ the Lagrange multiplier at instant i and $u_i \in \mathbb{R}^m$ the control input. Moreover, the following constraints are used to enforce the respect of the initial state and the system model:

$$\begin{aligned} x_k - s_k &= 0 \\ f(s_i, u_i) - s_{i+1} &= 0 \quad i \in \{k, \dots, k + N - 1\} \end{aligned}$$

The Karush-Kuhn-Trucker (KKT) conditions must be considered in order to handle the first order optimality conditions, leading to the following formulation:

$$\nabla_y \mathcal{L}^k = 0$$

where \mathcal{L}^k is the lagrangian at instant k . Normally, the above equation should be solved at each sampling time. Considering that the new idea consists in distributing the optimization during the runtime of the process, the new updating law to be applied has the following structure:

$$\nabla_y \mathcal{L}^k + J^k(y) \cdot \Delta y = 0 \quad (3.1)$$

where $J^k(y)$ is a symmetric matrix approximating the Hessian $\nabla_y^2 \mathcal{L}^k$. The feedback phase starts when the new measurement (or estimation) x_k of the state is acquired. This is done by calculating the control to be applied immediately to the real system (Fig. 3.4). It is clearly shown that the computational burden in the preparation phase is much more important than the feedback one. This results in a reduced delay between the acquisition of the state and the computation of the feedback.

Some interesting points must be emphasized in this real time iteration method. The matrix $J^k(y)$ is completely independent of the state x_k . This allows the computation of its value (during the preparation phase) before the real acquisition of the new value of the state. Moreover, the major part of the computation of the lagrangian $\nabla_y \mathcal{L}^k$ can also be done regardless the value of the state x_k . In this way, the only task to be executed between the acquisition and the control input calculation is to solve the linear system defined in (3.1). The particular structure of $J^k(y)$ may facilitate the resolution of such a system.

It goes without saying however that due to the choice of the decision variable that include the sequences of control, state and lagrange multipliers, the real-time iteration based on the multiple shooting approach leads to an optimization problem which is of extremely high dimension. Although the resulting problem is highly structured and can therefore be solved using sparse matrices related softwares, it remains true that the memory and computational requirements may be incompatible with the on-board available tools in a real mechatronic system. Another drawback of this approach is the unavoidable use of derivatives that may be cumbersome to compute.

3.2.2.2 Interior Point-based Methods

Despite the fact that high-dimensional models imply a very challenging problem for NMPC, some optimization strategies have been formulated to address such a challenging problem [Biegler and Zavala, 2009, Zavala and Biegler, 2009]. In [Biegler and Zavala, 2009], a very powerful algorithm called *IPOPT* [Wächter and Biegler, 2006] was successfully applied to large-scale industrial processes, containing more than 350 state variables and DAEs.

The *IPOPT* algorithm is based on the simultaneous collocation approach and uses the interior point method (also called barrier method) to treat NLP in a very efficient way. A logarithmic barrier method to inequality constraints in the NLP is used to define a sequence of constrained optimization problems. Such sequence is defined using a sequence of the coefficient of some barrier term μ in the cost function.

In the *IPOPT* method, the KKT conditions combined with the sparse Newton's method are both used to solve the sequence of problems and to assess stability conditions. In [Zavala et al., 2006], it is shown that the resulting controller has many advantages such as low computational cost to compute the exact Hessian and avoid expensive DAE integrations.

As in the real-time iteration scheme, the proposed strategy splits the optimization process into two main tasks for on-line dynamic optimization [Biegler and Zavala, 2009]: the background and on-line computation.

- **Background computation**

In this phase, the expensive computations are performed such as sensitivity, exact Hessian, decomposition and factorization steps. The dominant calculation is the repeated factorization of the large sparse matrix obtained from the KKT conditions together with the Newton's method, as mentioned above.

- **On-line computation**

This is the fast part of the algorithm, which uses the NLP sensitivity to calculate an estimated solution with perturbed initial conditions, allowing an immediate control injection. Then, once a measurement is obtained, the on-line update of the solution vector requires a single backsolve. This is one of the main part of the strategy, since this on-line procedure may take less than 1% of the dynamic optimization routine.

Therefore, the *IPOPT* method roughly shows the same advantages and drawbacks as the multiple shooting real-time iteration.

3.2.2.3 Continuation/Generalized Minimum Residual (C/GMRES)

The Generalized Minimum RESidual method (GMRES) has been proposed in [Ohtsuka, 2004]. In this method, the decision variable is the sequence of control and KKT multipliers gathered in a vector U . A differential equation is defined that must be met by U in order to enforce the respect of the necessary conditions of optimality. By doing so, there is no more, properly speaking, an optimization process but rather a dynamic state feedback in which the internal state is precisely U .

Let H be the Hamiltonian of the optimization problem to be solved:

$$H(x, \lambda, \mu, u) = L(x, u) + \lambda^T f(x, u) + \mu^T c(x, u)$$

where x is the state, $L(x, u)$ is the stage cost, $c(x, u)$ the equality constraints, μ the costate and λ the Lagrange multipliers associated with the equality constraint. The basic idea in the GMRES method is to try to meet the first order necessary conditions of optimality, namely $H_u = 0$ where U is the decision variable given by:

$$U = (u_0^T, \mu_0^T, \dots, u_{N-1}^T, \mu_{N-1}^T)^T$$

The values of x_i and λ_i for $i = \{0, \dots, N\}$ are obtained by solving respectively the system equations and the adjoint system equations $\dot{\lambda} = -H_x$ given the sequence U and the initial value x_0 of the state. Consequently, writing the first order optimality conditions lead to the following system of nonlinear equations:

$$F(U, x) = 0 = \begin{pmatrix} H_u(x_0, \lambda_0, \mu_0, u_0) \\ \vdots \\ H_u(x_{N-1}, \lambda_{N-1}, \mu_{N-1}, u_{N-1}) \end{pmatrix}$$

The idea is to find some dynamics for U that results in the convergence of U to the solution of $F(U, x) = 0$. This leads to the following stabilizing dynamics:

$$\dot{F}(U, x) = A_s \cdot F(U, x)$$

where A_s is a stable matrix. Therefore, assuming that F_U is nonsingular over the closed-loop system trajectory, the updating law for the decision variable U can be given by:

$$\dot{U} = F_U^{-1}(A_s \cdot F - F_x \cdot \dot{x}) \quad (3.2)$$

Note however that this assumes that one has already *good* initial guess for the value of U , otherwise, the problem is identical to the problem of solving high dimensional

system of nonlinear equations. The idea proposed in [Ohtsuka, 2004] is to find such an initial guess by a sort of continuation method (leading to the so-called C/GMRES method). The details related to the continuation related aspects are skipped here for the sake of simplicity.

Since integrating the differential system (3.2) involves expensive computations to evaluate gradient, the forward difference (FD) approximation for Jacobians is used together with the GMRES method [Kelley, 1995]. This results in a combined FD-GMRES approach. More precisely, the residual term r is defined by:

$$r = F_U \dot{U} + F_x \dot{x} - A_s F$$

Therefore, only the products $F_U \dot{U}$ and $F_x \dot{x}$ have to be evaluated. This can be done using finite difference approximation according to:

$$F_U W + F_x w \approx \frac{F(U + hW, x + hw) - F(U, x)}{h} \quad (3.3)$$

where h is a positive real number. Expression (3.3) clearly shows that the computational effort to perform the optimization process reduces drastically when compared to the calculation of F_U in (3.2), given the high dimension of the vector U .

Note however that the C/GMRES formulation remains a direct approach and hence presents the same risk of system integration instability when bad sequences of control input are encountered during the transient closed-loop trajectory. In order to improve the numerical accuracy and stability, a combination of GMRES and multiple shooting approach has been investigated [Shimizu et al., 2006].

3.2.2.4 Hybrid-Parameterized Scheme

The approach presented by [DeHaan and Guay, 2007] shares with the last method the differential character of the updating scheme. Indeed, several parameters are defined for the open loop trajectory, then a differential system describing the evolution of these parameters is derived in order to enforce the convergence towards the respect of optimality conditions. One *singular* feature of the approach proposed in [DeHaan and Guay, 2007] is that the decision instants are part of the decision variables vector. This gives the approach a somehow hybrid character.

The main drawback of this approach is that it assumes already known a system trajectory that meets the final constraint on the state needed to guarantee the feasibility and the stability of the closed-loop system. In some sense, this makes the proposed solution convenient to improve the performance of an already stabilized system (by another existing feedback law).

3.3 The Parameterized NMPC

3.3.1 Motivations

In this section, the low-dimensional parameterized NMPC scheme for fast systems, introduced in [Alamir, 2006a] is presented. This approach is particularly adapted to real-world context where the computation time is reduced but also where the on-board computational facilities and memory may prevent the use of oversophisticated formulations. The philosophy that underlines the parametrization approach lies in the following ideas:

1. Open-loop control profiles showing very simple time structures, when used in a receding-horizon framework, generally lead to very rich closed-loop control profiles that correspond to a small drop in the overall resulting optimality. This simple fact explains how, in the early years, MPC approaches have solved many industrial problems using only constant control over the prediction horizon.
2. Although the global optimum of the NMPC cost function corresponding to a low dimensional parametrization is necessarily higher than that of a classical trivial piecewise constant parametrization, for a well designed parametrization however, it is more likely that the former would be easier to achieve than the latter due to the difference in the problem complexity. More clearly, in a constrained computational context, the suboptimal solution of a simple optimization problem may be better than the suboptimal solution of a far more complex one. This simple idea is schematically depicted on Figure 3.5.
3. Classical piece-wise constant control parametrization in which all the control values are degrees of freedom result in unnecessarily high dimensional optimization problems. This is *a fortiori* true in the case where even the state values along the future system's trajectory are taken as degrees of freedom. Although these high dimensional problems are highly structured and can therefore be quite efficiently handled by dedicated algorithms, they still need incompressible high demanding preparation steps. Moreover, the dedicated softwares and memory storage needed for such problems are generally incompatible with embedded capacities. This is particularly true in the mechatronic and robotic applications.

There are many other advantages of the parameterized NMPC approach such as its ability to explicitly exploit the model structure that is generally strong in electromechanical systems and the possibility to use control parametrization that takes into account the constraints of the problem at the very definition of the parametrization.

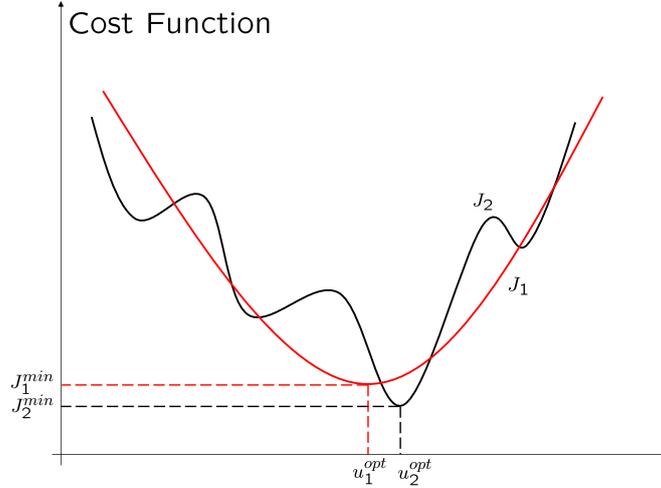


Figure 3.5: Schematic view showing how an approximate but well posed optimization problem (red line) may be worth using than an exact but highly accidental optimization problem. The global minima J_2^{min} may be quite difficult to obtain and the corresponding control value u_2^{opt} as well. On the other hand, u_1^{opt} can be easily obtained since J_1^{min} is the minimum of a simple and well-posed optimization problem.

Table 3.1 shows some examples giving the computation time needed to solve the optimization problem underlying the parameterized NMPC when applied to several examples of mechatronic systems. These examples are detailed in [Alamir, 2006a] except for the automated manual transmission example for which the reader can refer to [Amari et al., 2009, 2008]. The term n_p denotes the number of parameters used as degrees of freedom of the decision variable.

3.3.2 Definitions and Notation

This section introduces some definitions and notations related to the parameterized NMPC approach. First, let us consider a time-invariant dynamic model given in the following general form:

$$x(t) = X(t, x_0, \mathbf{u}) \quad (3.4)$$

where $x(t)$ is the state $x \in \mathbb{R}^n$ at instant $t \leq T$ when initialized at $(0, x_0)$ and under the control profile \mathbf{u} defined on $[0, T]$. Let $\tau_s > 0$ denote some sampling period such that $T = N\tau_s$.

Problem	n_p	Computation Time (ms)
Twin-pendulum system	1	< 5
Diesel engine air path	2	< 10
Automated Manual Transmission	1	< 0.5
Minimum interception time	1	< 20
Stabilization of the chained syst.	1	< 1
Stabilization of PVTOL	2	< 1
Stabilization of deficient satellite	2	< 10

Table 3.1: Some computation time examples for the successful use of parameterized NMPC in robotics and mechatronic [Alamir, 2006a, Amari et al., 2009, 2008]

Let us define the set of parameters p such that $p \in \mathbb{P}$, where $\mathbb{P} \in \mathbb{R}^{n_p}$ is some admissible subset of some convenient space. Then, each map $\mathcal{U}_{pwc} : \mathbb{P} \times \mathbb{R}^n \rightarrow \mathbb{U}^N$ defines on $[0, T]$ a parameterized piecewise constant (PWC) control profile (with parameters in $\mathbb{P} \times \mathbb{R}^n \subset \mathbb{R}^{n_p} \times \mathbb{R}^n$) such that:

$$\begin{aligned} \mathbf{u}(t) &= u^{(k)}(p, x) \quad ; \quad t \in [t_{k-1}, t_k] \quad ; \quad t_k = k\tau_s \\ \mathcal{U}_{pwc}(p, x) &:= (u^{(1)}(p, x) \quad \dots \quad u^{(N)}(p, x)) \in \mathbb{U}^N \end{aligned}$$

The state trajectory under the PWC control profile $\mathcal{U}_{pwc}(p, x_0)$ is denoted hereafter by $X(\cdot, x_0, p)$. More generally, using a straightforward abuse of notation, for each sampling instant $t_j = j\tau_s$ ($j \in \mathbb{N}$), the notation $X(t, x(t_j), p)$ denotes the state trajectory of the model at instant $t_j + t$ under the PWC control profile defined by $\mathcal{U}_{pwc}(p, x(t_j))$ over the time interval $[t_j, t_j + T]$. Chapter 2 showed that the NMPC strategy relies on the solution at each decision instant t_j of some optimization problem. In the parameterized context, one must be rewritten in the following form:

$$\hat{p}(x(t_j)) := \arg \min_{p \in \mathbb{P}} [J(p, x(t_j))] \quad \text{under} \quad C(p, x(t_j)) \leq 0 \quad (3.5)$$

where $J(p, x(t_j))$ is some cost function defined on the system trajectory starting from the initial condition $(t_j, x(t_j))$ under the p.w.c control profile defined by $\mathcal{U}_{pwc}(p, x(t_j))$. The condition $C(p, x(t_j)) \leq 0$ gathers all the problem constraints defined on the same trajectory including possible final constraints on the state.

Classical NMPC formulation states that once a solution $\hat{p}(x(t_j))$ is obtained, the first control in the corresponding optimal sequence $\mathcal{U}_{pwc}(p, x(t_j))$, namely $K(x(t_j)) := u^{(1)}(\hat{p}(x(t_j)), x(t_j))$ is applied to the system during the sampling period $[t_j, t_{j+1}]$. As shown in the previous chapter, this clearly results in the sampled-time state feedback law defined by:

$$K := u^{(1)}(\hat{p}(\cdot), \cdot) : \mathbb{R}^n \rightarrow \mathbb{U} \quad (3.6)$$

When a system with fast dynamic is considered however, only a finite number $i_t \in \mathbb{N}$ of iterations of some optimization process \mathcal{S} can be performed during the sampling period $[t_{j-1}, t_j]$. This lead to the following extended dynamic closed-loop system:

$$x(t_{j+1}) = X(\tau_s, x(t_j), p(t_j)) \quad (3.7)$$

$$p(t_{j+1}) = \mathcal{S}^{i_t}(p^+(t_j), x(t_j)) \quad (3.8)$$

where \mathcal{S}^{i_t} denotes successive finite number of iterations i_t of \mathcal{S} starting from the initial guess $p^+(t_j)$ which is related to $p(t_j)$ to guarantee (if possible) the translatability property (see [Alamir, 2006a]). This property makes it possible to construct a new control sequence (after the prediction horizon is shifted) that is built up with the remaining part of the past control sequence being used completed with a convenient final control.

The stability of the extended system (3.7)-(3.8) heavily depends on the performance of the optimizer \mathcal{S} , the number of iterations i_t and the quality of the model. The details about the stability issue can be seen in [Alamir, 2008].

The main drawback of the parameterized approach lies in the fact that there is no universal parametrization that can be applied to any system. The choice of the kind of parametrization is rather problem dependent. In a sense, this makes this approach more a way of thinking rather than a systematic design strategy. In this PhD work, this strategy is applied to solve two specific problems that are quite challenging from a computational point of view. This is the aim of the forthcoming chapters.

3.4 Conclusion

This chapter presented the main NMPC strategies developed to deal with fast dynamic systems. Some standard techniques, such as Real-Time Iteration scheme or Interior-Point method for example, have the interesting feature of being general methods to solve NLP to address the real-time problem of fast systems. On the other hand, such methods have to deal with sparse and high-dimension matrices. Moreover, the whole control sequence vector is taken as decision variable, and in some cases even the state vector, leading to a huge optimization problem. Despite the fact that the available tools may be used to more efficiently solve such problems, real-time implementation may be an issue in the cases where computational resources and available memory are limited.

In order to avoid these drawbacks, the parameterized NMPC scheme arises as an interesting alternative for fast systems control design. A well-structured parametrization scheme may lead to a low-dimensional optimization problem which is a quite important feature to handle real-time requirements. This is particularly true for a certain class of systems, such as those encountered in mechatronic and robotics for example. Since this PhD work focuses on the NMPC control design in such areas, in the forthcoming chapters, a parameterized NMPC strategy is presented for two practical applications: the diesel engine air path and the twin-pendulum system.

Chapter 4

NMPC for a Diesel Engine Air Path

4.1 Introduction

This chapter presents the formulation of a parameterized NMPC scheme, described in the previous chapter, to be applied on a diesel engine air path system. Diesel engines represent an interesting platform for predictive control application since they are constrained and highly nonlinear systems. One interesting feature of the proposed controller is that it uses no structural properties of the system model. Therefore, the proposed NMPC scheme can be applied to any future improved candidate model for diesel engines.

This chapter is organized as follows. First, a brief state of the art of the diesel engine air path control problem is provided in section 4.2. Important aspects concerning diesel engines are addressed such as the description of the process, the control problem, the existing solutions and the system modeling. Section 4.3 presents the observer design used for the estimation of the state vector of the model. Section 4.4 describes the parameterized NMPC approach for the control problem and shows how it can be used as a black-box solution for such processes. In section 4.5, a new *SQP* procedure involving a trust region mechanism is described and used to perform the optimization task. Finally, sections 4.6 and 4.7 present successively the simulation and experimental results. The latter have been obtained at the Johannes Kepler University, Austria. The chapter ends with a conclusion that summarizes the contribution and gives some guidelines for further investigations.

4.2 Diesel Engine

4.2.1 General Aspects

In the recent years, the automotive industry have witnessed a considerable development of the diesel engine passenger cars' production. The interest in this kind of vehicle does not stop growing thanks to the several advantages that are explained in this chapter. Figure 4.1 illustrates a BMW Diesel engine.



Figure 4.1: BMW Diesel engine.

In Europe, diesel engine vehicles have attracted the market's attention. According to the European Automobile Manufacturers Association (ACEA), the market share of new diesel car registrations in the western Europe quadrupled between 1990 and 2008, as shown in figure 4.2. The same study showed other impressive numbers for some countries. In France and Belgium for instance, both started at 33% in 1990 and climbed up to 77% and 79% in 2008, respectively. For Finland, the numbers are even more impressive. The market share climbed up from 5% to 50% in the same period.

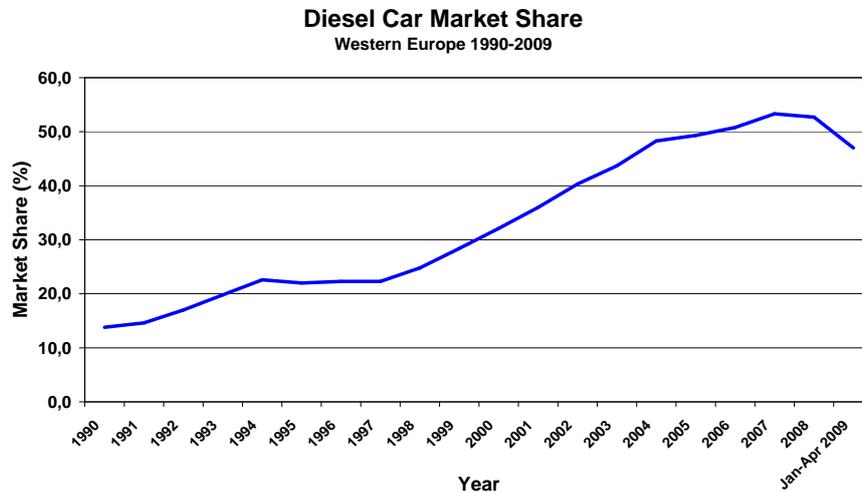


Figure 4.2: Diesel car market share in western Europe between 1990 and 2008.

In fact, it was not by chance that diesel engine vehicles have conquered this important segment of the automotive market. These engines have important features that are listed below [Heywood, 1988]:

- **Torque**

When compared to gasoline engines, diesel engines develop more torque and power at lower speeds. This is in particular due to the combustion process that takes place in such engines, also called, Compression-Ignition (CI) engines, instead of spark-ignition (SI) engines. In CI-engines, only the air is compressed to support higher pressures and temperatures, and when fuel is injected, it ignites automatically and no spark is needed. This means that diesel engines can work with higher compression ratios than gasoline engines, making the piston in the diesel engine travel further. Therefore, more torque can be delivered, since more work is extracted during the piston stroke.

- **Efficiency**

Actually, the diesel engine is the most efficient internal combustion engine yet available. The CI-engines have several interesting features that contribute directly to the engine's efficiency. First, a higher mechanical efficiency is provided since no throttle in the intake air is needed, eliminating flow losses in the intake part. On the other hand, a quite nice fuel conversion efficiency is

observed because CI-engines operate always lean so that the ratio of specific heats is higher over the expansion process. And finally, since there are no flow losses at throttle and the fresh charge does not contain any fuel, a higher volumetric efficiency can be reached for such engines.

- **Durability**

Since diesel engines must be designed and manufactured heavier than a gasoline engines to withstand the pressure within the engine, it can be expected to run many hours longer than its gasoline counterpart. Moreover, maintenance intervals are also generally longer for diesel engines.

As a matter of fact, diesel engines show many advantages over gasoline engines. Roughly speaking, CI-engines can add up to as much as 20% more efficiency than an SI-engine with similar size, with respect to fuel consumption. However, diesel engines have important drawbacks that must not be ignored. They are described in the sequel:

- **Emissions**

The emission problem is one of the main drawbacks of diesel engines [Johnson, 2001]. The pollutants elements consist of unburnt hydrocarbons (uHC), oxides of nitrogen, NO and NO₂, normally referred to as NO_x, carbon monoxide CO, and particulate matter PM, mainly soot. Comparing to the gasoline engines process, diesel combustion produces more NO_x and PM. On the other hand, CO emissions are negligible in CI-engines due to lean operation and emissions of uHC can be handled with oxidation catalysts. For this reason, the emissions of NO_x and PM deserves a special attention when diesel engines are considered.

- **Noise**

The noise generated by the combustion process in diesel engines represents another disadvantage of such engines. This comes from the initial rapid heat release immediately following the ignition-delay period where a significant quantity of premixed charge exists. To overcome this problem, a small amount of fuel prior to the main injection, which smoothes the heat release, must be injected. This method can be adapted to the more recent and modern injection systems. Another way to tackle the noise problem is to apply an injection rate modulation with a modern common rail injection system to achieve lower combustion noise. This makes the noise generated by diesel engine almost the same as the gasoline one, once it has warmed up.

With the drawbacks listed above, the formation of emissions, namely NO_x and PM, must be analyzed in details and they are explained in the sequel.

4.2.2 Formation of Emissions

The major part of NO from the diesel combustion is formed by the oxidation of atmospheric nitrogen. Actually, the formation of NO is highly dependent on the temperature since the forward reaction $N_2 + O \rightarrow NO + N$ and the reverse reactions $N + O_2 \rightarrow NO + O$ and $N + OH \rightarrow NO + H$ require a huge activation energy [Heywood, 1988]. Such reactions start at temperatures over than 1900 K and the NO stays around during cooling, since the reverse reaction is very slow. Therefore, the production of NO₂ can be obtained from NO through the reaction $NO + HO_2 \rightarrow NO_2 + OH$. However, the formed NO₂ is converted back to NO via $NO_2 + O \rightarrow NO + O_2$.

In fact, according to the diesel combustion process, important temperatures in a region where oxygen is available, make such area the ideal region for the formation of NO_x. In [Flynn et al., 1999], it is shown that two-thirds of the total NO_x emissions are formed in the diffusion flame while one-third is formed in the hot post-combustion gas regions. For SI-engines, the ratio NO₂/NO can be neglected thanks to the typical behavior of the flame temperature of the combustion process. Figure 4.3 illustrates the typical temperature and chemistry profile of the combustion flame.

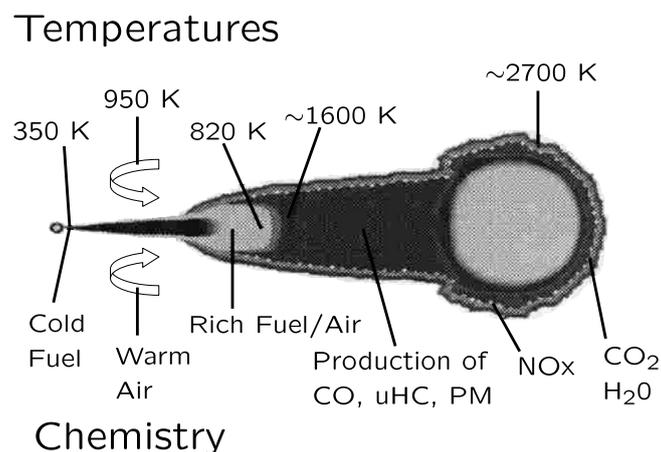


Figure 4.3: Schematic view of the temperature and chemistry of a typical diesel engine flame [Flynn et al., 1999].

There is another important pollutant emission, the soot particles referred as PM, which is formed in the cylinder in the regions where the combustion are not homo-

geneous. They heavily contribute to the total mass of particulate matter emitted by the engine. In order to avoid PM, it is quite interesting to work with high temperatures to burn-up the soot at the boundary of the diffusion flame sheath. On the other hand, high temperatures may lead to the formation of NOx. This paradox is also called NOx-PM trade-off, as shown in figure 4.4 together with the maximum admissible values according to the European norms Euro III, IV and V.

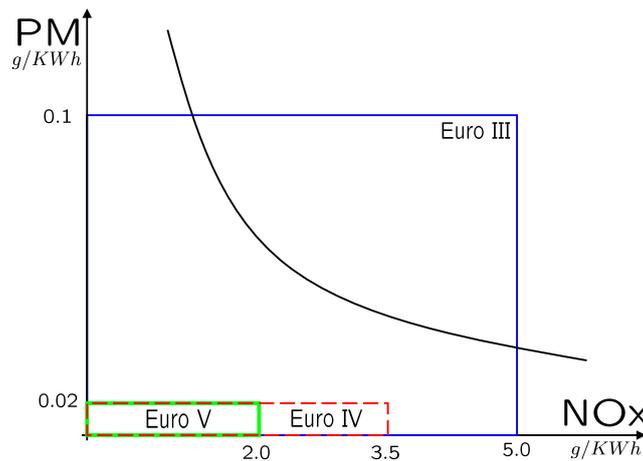


Figure 4.4: Schematic view of the trade-off between NOx and PM for diesel vehicle.

4.2.3 The Control Problem

In order to reduce the diesel engine emissions, it is quite important to understand the air path process of the engine to formulate the control problem to be solved. Figure 4.5 shows the complete scheme of the diesel engine air path, equipped with a turbo charger.

The air path works as follows: The compressor pumps the fresh air into the inlet manifold of the engine in order to boost the pressure. The fuel is directly sprayed into the combustion chamber and is burnt with the delivered air coming from the compressor. A part of the exhaust gas of the exhaust manifold is recirculated into the inlet manifold by means of the Exhaust Gas Recirculation (EGR) valve, which is extremely important to reduce the NOx emission. An inter cooler is used to decrease the fresh air temperature and EGR cooler to reduce the recirculated gas temperature. The remaining gas that is not recirculated goes to the Variable Geometry Turbine (VGT), which absorbs the heat energy from the exhaust gas and propels the compressor, which pumps the fresh air and closes the whole cycle.

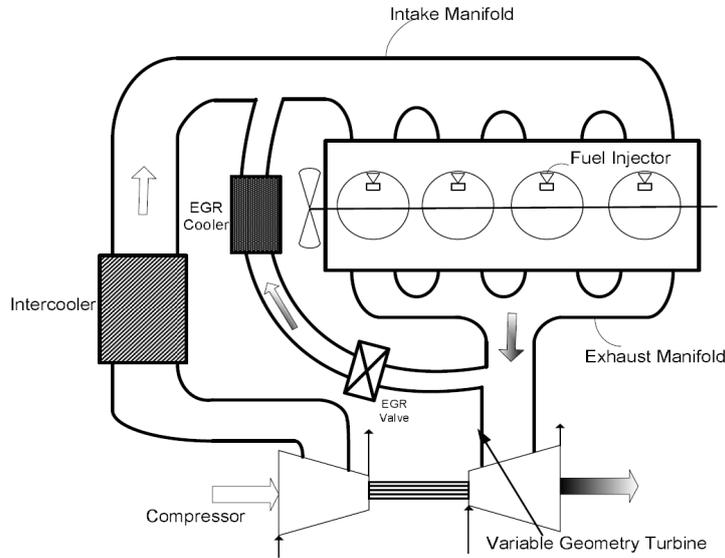


Figure 4.5: Schematic view of the complete diesel engine air path [Wei, 2006].

The EGR and VGT valves play a crucial role in the emission and the air path control [van Nieuwstadt et al., 2000]. The EGR reduces the oxygen fuel ratio during the combustion process, since it dilutes the incoming fresh air. Therefore, the flame temperature gets low in the diffusion flame and in the premixed burning. As seen previously, low temperature can attenuate the formation of NO_x (but increase the PM according to the NO_x-PM trade-off). Moreover, the EGR also raises the inlet manifold gas temperature, since the exhaust gas is much hotter than the one in the inlet manifold. For this reason, EGR gas must be cooled before it comes back onto the engine, in order to minimize the inlet charge temperature effect. Another effect of the EGR is that it delays the ignition time, but this can be compensated for by adjusting the injection timing. The details about all effects of EGR in the combustion process can be seen in [Jacobs et al., 2003].

On the other hand, the VGT valve has the important role of increasing the power-density of the engine, by forcing more air into the cylinder so that more fuel can be burnt in the same volume without increasing the air fuel ratio. For this reason, such engines are also called turbo-charged engines. This enables one to deal with the limitation of the amount of fuel that can be injected into the cylinder.

Once the process is described, the control problem can be formalized. The control inputs for the diesel engine air path are the positions of the EGR and VGT valves ($u \in \mathbb{R}^2$, in %). The engine speed N_e (in RPM), and fuel rate w_f (in mg/stroke), are the external inputs, which can be regarded as measured disturbances $w \in \mathbb{R}^2$.

In this PhD work, the fuel injection problem is not considered, but some important contributions are presented in [Dupraz, 1998, Gauthier, 2007].

It is worth underlying however that the output $y \in \mathbb{R}^2$ of the system that is used for the control design is not composed of the emissions, but the mass air flow rate MAF into the intake manifold, $y_1 = W_{ci}$ (in kg/h), and the intake manifold absolute pressure MAP, $y_2 = p_i$ (in kPa). The reason for using MAF and MAP as regulated outputs is that the engine is equipped with sensors for such signals while there are no on-board sensors for the emissions in commercial vehicles. The reference values for these regulated output are optimized in steady-state with respect to emissions and are mapped over the entire speed and fuel profiles. Therefore, provided that such set-points are correctly chosen, a good tracking of these variables may lead to an acceptable level of emissions. The choice and generation of the set points of MAP and MAF to cope with emissions is a challenging problem and more details can be found in [van Nieuwstadt et al., 2000]. In this work, it is assumed that these values are correctly chosen and hence, the quality of the control is estimated with respect to this *auxiliary* problem.

The control task is therefore to adjust the VGT and EGR valves positions to track the required boost pressure MAP and the air mass flow rate MAF set-points. More formally, the control problem is to design a controller that forces y to track some desired set-point y^d , such that:

$$y^d = [y_1^d, y_2^d]$$

Moreover, the inputs u must satisfy the following set of constraints:

$$u \in [u_{min}, u_{max}] \quad ; \quad u_{min} \in \mathbb{R}^2 \quad ; \quad u_{max} \in \mathbb{R}^2 \quad (4.1)$$

$$\delta u \in [-\delta_{max}, +\delta_{max}] \quad ; \quad \delta_{min} \in \mathbb{R}^2 \quad ; \quad \delta_{max} \in \mathbb{R}^2 \quad (4.2)$$

where the inclusions are to be considered componentwise and where δu are the increments on the inputs and can be defined as $\delta u(k) = u(k) - u(k-1)$. The next section describes some existing solutions that have been proposed to tackle the above control problem.

4.2.4 Existing Control Strategies

Several control strategies have been proposed to address the above problem. From a control point of view, the diesel engine is a highly coupled multi-variable system due to the combined effects of VGT and EGR [Nieuwstadt et al., 1998]. Moreover, the system is quite nonlinear (which is shown in the next section) and constrained.

Therefore, sophisticated controllers are necessary to correctly address the underlying control problem.

In [Jankovic and Kolmanovsky, 1998], a classical Lyapunov-like approach coupled with a feedback linearization technique has been used in order to design a nonlinear robust controller. Other nonlinear-like strategies proposed by [Plianos and Stobart, 2007, Chauvin et al., 2006, 2008] were developed based on the flatness property introduced by [Fliess et al., 1992]. Another family of works is based on linearization techniques or Linear Parameter Varying (LPV) models in order to use the advanced tools that are available for such models [Jung, 2003, Wei, 2006].

However, from a conceptual point of view, the presence of constraints on the control inputs and nonlinearities in the system model make predictive controllers an interesting solution for diesel engines. On the other hand, the most important drawback in such approaches concerns the real-time implementability. This is because finding the optimum solution may require a huge computation effort that may be incompatible with the available computation time. Therefore, in this example, one is in the heart of the context that makes appropriate the real-time dedicated formulations invoked in the preceding chapters.

Recently, some important contributions emerged that enable real-time implementation of linear MPC to diesel engines. The first important contribution concerning MPC for diesel engines has been presented in [Ortner and del Re, 2007]. This work uses the explicit MPC method [Bemporad et al., 2002], mentioned in the previous chapters, by using the Hybrid Toolbox software [Bemporad, 2004]. In this scenario, 12 linear models were identified, according to different values of injected fuel and engine speed. The Hybrid Toolbox provides the corresponding look-up table (off-line) that enables to associate a control gain to each region in the extended space represented by the state, set-point and control. Therefore, the only on-line task is the computation of the current region that is defined each by a set of linear affine in the extended state inequalities. However, the active region in this extended space is computed for the 12 models in parallel. The corresponding computational load explains the use of a control horizon of 1, with a sampling period of 50 ms. Indeed, authors mentioned the difficulty to work with longer control horizons given the available computation time.

The second important contribution involving predictive control has been proposed in [Ferreau et al., 2007], where the on-line active set strategy [Ferreau et al., 2006] is used. The main difference when compared to the previous approach is that the problem is solved on-line (using distributed in time optimization) to avoid important off-line computations. However, only 2 linear models corresponding to two different values of the engine speed were used to show the efficiency of the proposed strategy. Moreover, the control horizon was set to 5 and authors mentioned that increasing

this value is useless because this does not lead to an improvement of the tracking performance due to the modeling uncertainties. Authors also emphasized that more realistic models are needed in order to improve the performance.

In [Herceg et al., 2006], a third-order nonlinear model is used to design a NMPC controller in order to improve the transient behavior of a diesel engine. Here, the quasi-infinite horizon NMPC approach with guaranteed closed loop stability is used [Chen and Allgöwer, 1998] and simulations with a sequence of set-points were performed to compare the results with the linear and a nonlinear state feedback controller. However, this NMPC solution does not consider any particular algorithm to solve the resulting optimization problem in a faster way and hence, the proposed solution is not real-time implementable.

In this chapter, a parameterized NMPC approach is proposed that can be used even with nonlinear models. No assumptions on the model structure is needed nor any kind of linearization technique or multi-model framework. First, the system model is briefly described in the next section.

4.2.5 System Modeling

Diesel engine air path is a highly nonlinear system with many discontinuities and dead zones. In the literature, the Diesel engine behavior is normally represented by designing mean value models (MVM) for simulation and control design purposes. The parameters of such representation are derived by the steady state analysis of several components of the engine. However, the identification methods and a detailed description of the model are not in the scope of this PhD work, but it is quite evident that such study is one of the most important subjects related to diesel engines. Some important works concerning modeling and identification of diesel engines can be found in [Christen et al., 2001, Jung, 2003, Wei, 2006].

For the validation of the parameterized NMPC controller, two kinds of models are used in the sequel: a physical model and an identified model. The first model shows significant nonlinearities and involves several look-up tables. The second one, two identified models are proposed. The first is a higher order model which is linear up to an output injection and the second with an input injection. The identification process of both methods were developed using the BMW diesel engine test bench available at the the university of Linz¹. The parameters of the models are shown in appendix B and more details are presented in the following sections.

¹BMW M47TUE Diesel engine, at Johannes Kepler University Linz

4.2.5.1 Physical Model

The standard nonlinear model for diesel engines is used to describe the behavior of the engine. This nonlinear model has been identified directly from the diesel engine test bench and it is presented in [Langthaler, 2007]. The classical MVM for identification of all the parameters of the physical model is used. In addition, some variables such as efficiencies and some temperatures are modeled as polynomials with the following notation:

$$\mathcal{P}_i(v_1, v_2, \dots, v_{N_v})$$

where \mathcal{P}_i represents a second order polynomial identified on the diesel engine test bench depending on some set of variables $(v_1, v_2, \dots, v_{N_v})$. This leads to a look-up table obtained through interpolation. Figure 4.6 illustrates an example of a polynomial map. The values are not shown due to a confidential term with the University of Linz.

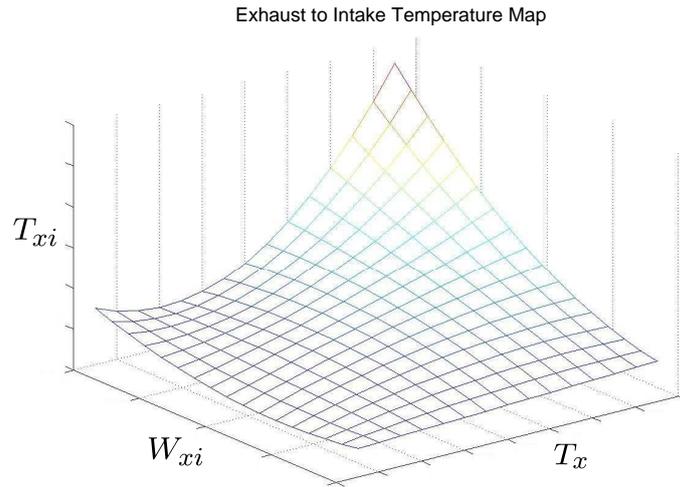


Figure 4.6: Polynomial map $T_{xi} = \mathcal{P}_1(T_x, W_{xi})$.

To better understand how the engine works, the later is split into four main parts, namely:

- Manifolds
- Exhaust Gas Recirculation
- Cylinder
- Turbocharger

Each of these parts of the engine has a specific model but they are highly coupled since there are several variables that are shared between them. These models are successively detailed.

Manifolds

The inlet manifold is directly linked to the combustion through the inlet valves. Here the compressor, which is driven by the turbine, propels the fresh air into the inlet manifold and part of the exhaust gas from the exhaust manifold is recirculated back to the inlet manifold, as already shown in figure 4.5. After the combustion process, the burned fuel and gas are pumped into the exhaust manifold. The exhaust gas has much higher temperature than the gas in the inlet manifold. The inlet manifold temperature is varying with the pressure and the gas mixture (air and exhaust gas). It can be obtained from the ideal gas equation:

$$\begin{aligned} T_i &= \frac{V_i}{Rm_i} p_i \\ T_x &= \frac{V_x}{Rm_x} p_x \end{aligned}$$

where R is ideal gas constant, $T_{(\cdot)}$, $V_{(\cdot)}$, $m_{(\cdot)}$ and $p_{(\cdot)}$ are the temperature, volume, mass and pressure respectively. The index i corresponds to the intake manifold and x , the exhaust manifold.

The exhaust manifold dynamics directly affects the turbine of the turbocharger system which receives the energy from the exhaust gas and drives the compressor to supply fresh air to the inlet manifold. The thermodynamic equations of the intake and exhaust manifold pressures can be expressed as follows:

$$\begin{aligned} \dot{p}_i &= \frac{R\kappa}{V_i} (W_{ci}T_e - W_{ie}T_i + W_{xi}T_{xif}) \\ \dot{p}_x &= \frac{R\kappa}{V_x} ((W_{ie} + W_f)T_e - (W_{xi} + W_{xt})T_x) \end{aligned}$$

where $W_{(\cdot)}$ is the mass flow rate and the index ci denotes compressor to intake, ie intake to engine, ex engine to exhaust, xi exhaust to intake and xt exhaust to turbine and κ the specific heat ratio.

The term T_{xif} represents the final value of the temperature from the exhaust to intake through the inter-cooler. Its dynamics can be represented by a first order filter as follows:

$$\dot{T}_{xif} = \frac{1}{\tau_{EGR}} (-T_{xif} + T_{xi})$$

where τ_{EGR} is the constant time of the EGR valve. Moreover, the T_{xi} is a function of T_x and W_{xi} that is generally represented by a polynomial expression. This leads to the following notation:

$$T_{xi} = \mathcal{P}_1(T_x, W_{xi})$$

To complete the manifold model, the mass balance in the intake and the exhaust can be written as follows:

$$\begin{aligned}\dot{m}_i &= W_{ci} - W_{ie} + W_{xi} \\ \dot{m}_x &= W_{ie} + W_f - W_{xi} - W_{xt}\end{aligned}$$

Exhaust Gas Recirculation

The EGR connects the exhaust with the intake manifold. Mass accumulation can be neglected because of the short distance and the direct connection. In the literature, the flow rate equation W_{xi} is given by:

$$W_{xi} = \begin{cases} \frac{-A(u_{egr})p_x}{\sqrt{RT_x}} \sqrt{2p_r(1-p_r)} & \text{if } p_x \geq p_i \\ \frac{-A(u_{egr})p_i}{\sqrt{RT_i}} \sqrt{\frac{2}{p_r} \left(1 - \frac{1}{p_r}\right)} & \text{if } p_i < p_x \end{cases}$$

where $p_r = p_i/p_x$ represents the pressure ratio and A is the effective area of EGR which depends on the valve position u_{egr} . This is the second identified map of the model, since A is a polynomial function of u_{egr} , such that:

$$A(u_{egr}) = \mathcal{P}_2(u_{egr})$$

Cylinder

The classical model of the cylinder is given by the equation of the mass flow rate, which connects the intake to engine

$$W_{ie} = \eta_v \frac{m_i}{V_i} \frac{N_e}{2\pi} \frac{V_d}{2}$$

where V_d represents the total engine displacement volume and η_v the volumetric efficiency that depends basically on two variables: the MAP (intake pressure) and the engine speed. For this reason, a third identified map for η_v can be written as follows:

$$\eta_v = \mathcal{P}_3(p_{in}, N_e)$$

Another important map to take into account is the engine to exhaust temperature T_{ex} . This temperature is influenced by the mass that flows into the cylinder and belonging temperatures, cylinder wall temperature and so on. Here, only the main variables are considered, namely, the fuel mass w_f and the mass flow rate from the intake manifold W_{ie} into the cylinder, leading to the following polynomial map:

$$T_{ex} = \mathcal{P}_4(w_f, W_{ie})$$

Turbocharger Model

For the turbocharger model, it is considered that the pressure and temperature after the turbine can be assumed to be a constant value and near to the ambient temperature respectively. Then, three important variables must be considered: the exhaust to turbine flow W_{xt} , the compressor to intake flow W_{ci} and the turbine power P_t . The W_{xt} basically depends on the turbocharger position u_{vgt} and the exhaust pressure p_x according to:

$$W_{xt} = \mathcal{P}_5(u_{vgt}, p_x)$$

Therefore, the turbine power P_t can be determined by:

$$P_t = W_{xt} c_p T_x \left(1 - \left(\frac{p_a}{p_x} \right)^\kappa \right)$$

where c_p represents the specific heat at constant pressure and p_a the ambient pressure. A dynamic equation corresponding to the power transmitted to the compressor denoted by P_c , can be easily determined by using the compressor efficiency η_m and the time constant τ_{vgt} for the opening of VGT. A first order system can be written as follows:

$$\dot{P}_c = \frac{1}{\tau_{vgt}} (-P_c + \eta_m P_t)$$

The resulting compressor flow W_{ci} to be injected in the intake manifold is a function of the compressor power P_c and the intake manifold pressure p_i , according to:

$$W_{ci} = \mathcal{P}_6(P_c, p_i)$$

In the same way, the temperature compressor to intake T_{ci} depends not only on the above mass flow rate, but also on the intake pressure:

$$T_{ci} = \mathcal{P}_7(W_{ci}, p_i)$$

This ends the set of nonlinear equations of the diesel engine air path process.

Summarizing the Engine Model

To summarize, the whole behavior of the diesel engine air path can be described by a sixth-order system, with 5 algebraic equations and 7 look-up tables.

Dynamic Equations

$$\begin{cases} \dot{p}_i = \frac{R\kappa}{V_i}(W_{ci}T_e - W_{ie}T_i + W_{xi}T_{xif}) \\ \dot{p}_x = \frac{R\kappa}{V_x}((W_{ie} + W_f)T_e - (W_{xi} + W_{xt})T_x) \\ \dot{P}_c = \frac{1}{\tau_{vgt}}(-P_c + \eta_m P_t) \\ \dot{m}_i = W_{ci} - W_{ie} + W_{xi} \\ \dot{m}_x = W_{ie} + W_f - W_{xi} + W_{xt} \\ \dot{T}_{xif} = \frac{1}{\tau_{egr}}(-T_{xif} + T_{xi}) \end{cases} \quad (4.3)$$

Algebraic Equations

$$\begin{cases} W_{xi} = \begin{cases} \frac{-A(u_{egr})p_x}{\sqrt{RT_x}} \sqrt{2p_r(1-p_r)} & \text{if } p_x \geq p_i \\ \frac{-A(u_{egr})p_i}{\sqrt{RT_i}} \sqrt{\frac{2}{p_r}\left(1 - \frac{1}{p_r}\right)} & \text{if } p_i < p_x \end{cases} \\ T_i = \frac{V_i}{Rm_i}p_i \\ T_x = \frac{V_x}{Rm_x}p_x \\ W_{ie} = \eta_v \frac{m_i}{V_i} \frac{N_e}{2\pi} \frac{V_d}{2} \\ P_t = W_{xt}c_p T_x \left(1 - \left(\frac{p_a}{p_x}\right)^\kappa\right) \end{cases} \quad (4.4)$$

Polynomial Maps / Look-up tables

$$\begin{cases} T_{xi} = \mathcal{P}_1(T_x, W_{xi}) \\ A(u_{egr}) = \mathcal{P}_2(u_{egr}) \\ \eta_v = \mathcal{P}_3(p_{in}, N_e) \\ T_{ex} = \mathcal{P}_4(w_f, W_{ie}) \\ W_{xt} = \mathcal{P}_5(u_{vgt}, p_x) \\ W_{ci} = \mathcal{P}_6(P_c, p_i) \\ T_{ci} = \mathcal{P}_7(W_{ci}, p_i) \end{cases} \quad (4.5)$$

It is quite clear that the system of equations (4.3)-(4.4)-(4.5) describing the diesel engine air path model is a highly nonlinear system. Moreover, recall that the inputs must respect the set of constraints (4.1)-(4.2).

4.2.5.2 Empirical Models

In this section, the empirical models that have been developed for the diesel engine test bench of the university of Linz are described. As mentioned previously, the first identified models for predictive control were represented by linearized systems around some operational point [Ortner and del Re, 2007, Ferreau et al., 2007]. However, since diesel engines show strong nonlinearities and active constraints on the inputs that have to be correctly handled, standard multi-linear methods may be insufficient to cover the whole engine operational range and the use of fully or appropriate nonlinear models may be an important issue. Control design methods that can cope with such models become therefore mandatory.

A data based identification process realized in [Ortner and del Re, 2007] using the test bench of the university of Linz, showed that the MAF and the MAP are basically influenced by the control inputs EGR, VGT, and by two measured disturbances, fuel injection w_f and engine speed N_e .

Figure 4.7 shows a simplified view of the diesel engine, with the definition of the main variables used for control design. In this test bench, the set-points of MAF and MAP are provided by a linear interpolation according to two-dimensional look-up table depending on the operational point defined by the fuel injection and the engine speed.

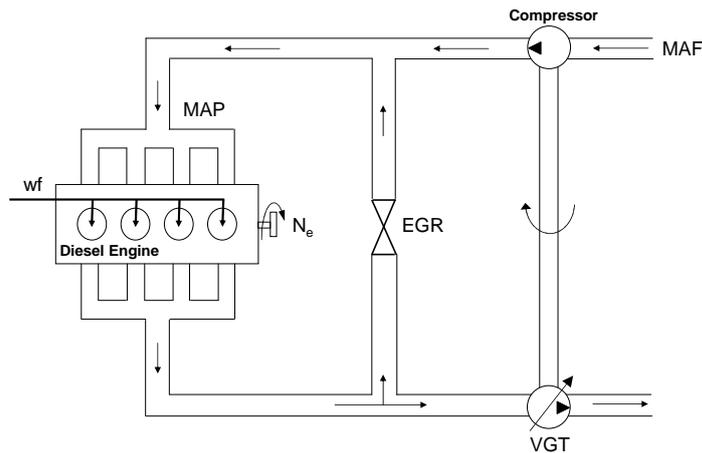


Figure 4.7: Schematic view of the simplified diesel engine air path and the variables used for control design.

Based on these considerations, two nonlinear models were developed and identified from the test bench: the first model is a linear model up to an output injection while the second is a linear model up to an input injection.

Nonlinear Model Depending on the Outputs

This model that involves a state vector of dimension 14 has been obtained using a sampling period of 10 ms. It shows the following structure:

$$\begin{cases} x^+ = [A(y - y_c)]x + B_1[u - u_c] + B_2[w - w_c] \\ y = [C(y - y_c)]x + y_c + \varepsilon \\ \varepsilon^+ = \varepsilon \end{cases} \quad (4.6)$$

where $x \in \mathbb{R}^n$ is the state ($n = 14$), y , u and w the vector of output, control and measured disturbances respectively. ε is the current estimation of the prediction error that enables uncertainties and model mismatch to be handled. u_c , y_c and w_c are the central values of input, output and disturbances respectively at the operation point used to identify the model. Consequently, the set of central values shifts the whole system to work around 0. The matrices $B_1 \in \mathbb{R}^{14 \times 2}$ and $B_2 \in \mathbb{R}^{14 \times 2}$ are constant while $A(\cdot) \in \mathbb{R}^{14 \times 14}$ and $C(\cdot) \in \mathbb{R}^{2 \times 14}$ depend on the output vector y and were identified at a set of central values y_c . This clearly makes the present model nonlinear. However, the pair $[A(\cdot), C(\cdot)]$ is not observable, and this will be discussed later.

Note that this class of model is widely used. In this case, instead of considering a multi-model approach typically found in linearized strategies, a more realistic viewpoint is adopted by keeping in mind the continuous nature of the dependence with respect to the variable y .

This model was the first one used for testing the NMPC controller and only simulation results were performed, which are presented in section 4.6. The experimental results were performed with the model dependent on the inputs described in the next topic.

Nonlinear Model Depending on the Inputs

The second model [Ortner et al., 2009] shows the same structure than before except that the system matrices depend on the input rather than on the output. The resulting model involves 8 states and has been obtained using a sampling period of 50 ms. More precisely:

$$\begin{cases} x^+ = [A(u - u_c, w - w_c)]x + B_1[u - u_c] + B_2[w - w_c] \\ y = [C(u - u_c, w - w_c)]x + y_c + \varepsilon \\ \varepsilon^+ = \varepsilon \end{cases} \quad (4.7)$$

where $x \in \mathbb{R}^n$ is the state ($n=8$). The matrices B_1 and B_2 are constant while $A(\cdot)$ and $C(\cdot)$ are now dependent on inputs u and w instead of y .

One of the advantages of (4.7) when compared to the output dependent model (4.6) lies in the fact that the pair $[A(\cdot), C(\cdot)]$ is observable for each value of the control in the operational region of interest. Moreover, it leads to more accurate results. For this reason, model (4.7) was chosen by our partners of the university of Linz for the experimental validation. On the other hand, the model (4.6) corresponds to a sampling rate of 10 ms compared to 50 ms for the model given by (4.7). This would have been interesting in order to enable a more reactive controller to be obtained.

4.2.5.3 Some Considerations About the Models

As mentioned above, the outputs, measurement disturbances and the control inputs were identified around some central values, namely, y_c , w_c and u_c , respectively. This is only valid for the empirical models (4.6) and (4.7) since the full nonlinear model (4.3)-(4.5) does not depend on the central values. Nevertheless, some few considerations must be done concerning these values.

For the sake of simplicity, the variations on y and w will be made around 0 and hence the notations y_c and w_c will be omitted hereafter. Indeed, this simplification is also valid for experimental validation since the central values are managed by an external loop, and hence the values of y_c and w_c can be internally set to 0.

However, for the increments on the input vector, the central values u_c must be considered in order to respect the constraints on the inputs, as shown in (4.1). In fact, for models (4.6) and (4.7), the inputs u must satisfy:

$$u + u_c \in [u_{min}, u_{max}] \quad ; \quad u_{min} \in \mathbb{R}^2 \quad ; \quad u_{max} \in \mathbb{R}^2$$

Therefore, the above expression correctly defines the constraints on the inputs to be respected for models (4.6) and (4.7). Nevertheless, in order to unify and simplify notations, for further developments involving the computation of the control input u , the term u_c will be also omitted.

In the forthcoming sections, several approaches and scenarios are shown concerning the models presented in this topic. In order to clarify and organize this dissertation,

table 4.1 shows the link between the model and the corresponding sections that are presented in the sequel.

Table 4.1: Sections where the models are referred

Model	Observer	Controller	Simulations	Experiments
Depending on the outputs (4.6)	(4.3)	(4.4.3.1)-(4.4.4)	(4.6.1)	-
Depending on the inputs (4.7)	(4.3)	(4.4.3.2)-(4.4.4)	(4.6.2)	(4.7)
Full nonlinear (4.3)-(4.5)	-	(4.4.5)	(4.6.3)	-

4.3 Observer Design

4.3.1 The Relevance of the Observer

The controller to be implemented needs the states that affect the evolution of the regulated output to be reconstructed. In the physical model presented in (4.3), the most important component of the state are measured. The only components that need to be reconstructed are the states of the first order filters for T_{xif} and P_t . This task is quite easy to implement.

On the other hand, the data-based models (4.6) and (4.7) are not that simple since many of the identified states have no physical meaning and hence, they can not be measured. In such cases, designing an observer becomes essential to recover the state.

The first nonlinear model (4.6) depending on the outputs is not observable. Chronologically, this model was the first delivered by our partners from the Johannes Kepler University in order to proceed with a preliminary study of the parameterized NMPC solution. Due to this lack of observability, standard techniques, such as Extended Kalman Filter (EKF), for observer design could not be correctly applied. For this reason, a nonlinear Moving Horizon Observer (MHO) was proposed and it is presented in this section. Note that rigorously speaking, there is no algorithm that can be called *observer* for a non observable system. The term observer here refers to an algorithm that re-construct the observable part of the state vector.

The second nonlinear model (4.7) with input related non linearities is observable for any input vector belonging to the set of operational values of interest. This model was the second one developed by our partners and is used hereafter for the experimental validation. For this reason, the structure of the MHO, initially developed for the first model (4.6), was also adopted to this one, since the affine structure is

the same and at instant k , all the inputs are known. The next topic goes further into the MHO design.

4.3.2 Moving Horizon Observer Design

The MHO is the dual paradigm of the predictive control and basically have the same advantages and drawbacks of NMPC [Michalska and Q.Mayne, 1995, Alamir and Calvillo-Corona, 2002, Alamir, 1999]. These observers can deal with nonlinearities and constraints in the system but also need on-line optimization to be performed that can be incompatible with the available computation time.

Basically, the main idea of MHO's is to estimate the current state using a moving and fixed-size window containing N_o past measurements. The oldest measurements are discarded when the newest one become available. Then, the collected data are used to define a cost function, which is minimized with the earliest state as a decision variable. As mentioned previously, the observer will be developed using the model given by (4.7) since this is the model that was used in the real implementation, but the principle is the same for designing an observer for model (4.6).

Let k be the current instant, \bar{Y}_k and \bar{U}_k the vectors of past output measurements and input measurements respectively, defined according to:

$$\bar{Y}_k = \begin{pmatrix} y(k - N_o + 1) \\ \vdots \\ y(k) \end{pmatrix} \in \mathbb{R}^{m \cdot N_o} ; \bar{U}_k = \begin{pmatrix} u(k - N_o + 1) \\ w(k - N_o + 1) \\ \vdots \\ u(k) \\ w(k) \end{pmatrix} \in \mathbb{R}^{2m \cdot N_o}$$

Moreover, considering the error ε due to the wrong prediction, the vector \bar{E}_k of the past values of ε becomes:

$$\bar{E}_k = \begin{pmatrix} \varepsilon(k - N_o + 1) \\ \vdots \\ \varepsilon(k) \end{pmatrix} \in \mathbb{R}^{m \cdot N_o}$$

The value of ε is updated at each sampling instant k according to:

$$\varepsilon(k) = \varepsilon(k - 1) + k_i \cdot (y^p(k) - y^m(k)) \quad (4.8)$$

where $y^p(k)$ and $y^m(k)$ are, respectively, the predicted value under the previous estimated state at instant $k - 1$ and the measured output at instant k , $\varepsilon(k - 1)$ the previous value of the prediction error and $k_i \in \mathbb{R}^m$ the integrator gain.

Therefore, straightforward computations lead to:

$$x(k) = [\Phi(\bar{U}_k)] \cdot x^{(-)} + [\Psi(\bar{U}_k)] \cdot \bar{U}_k \quad (4.9)$$

$$\bar{Y}_k = [\Omega(\bar{U}_k)] \cdot x^{(-)} + [\Gamma(\bar{U}_k)] \cdot \bar{U}_k + \bar{E}_k \quad (4.10)$$

where k is the current instant, $x^{(-)}$ the state at instant $k - N_o + 1$. The matrices $\Phi(\bar{U}_k)$, $\Omega(\bar{U}_k)$, $\Gamma(\bar{U}_k)$ and $\Psi(\bar{U}_k)$ are obtained after straightforward computations in terms of the system matrices $A(\cdot)$, $C(\cdot)$, $B = [B_1 \ B_2]$. The details of all these calculations are given in the appendix A.

Based on the above expressions, one can argue that if the past estimation \hat{x}_{k-1} and the value of $x^{(-)}$ were correct, one should have the following two equalities

$$\begin{aligned} A(u(k-1), w(k-1))\hat{x}(k-1) + B[u(k-1) \ w(k-1)]^T &= [\Phi(\bar{U}_k)] \cdot x^{(-)} + [\Psi(\bar{U}_k)] \cdot \bar{U}_k \\ \bar{Y}_k &= [\Omega(\bar{U}_k)] \cdot x^{(-)} + [\Gamma(\bar{U}_k)] \cdot \bar{U}_k + \bar{E}_k \end{aligned}$$

Note that these two equations can be used to define conditions on the past value $x^{(-)}$ that one has to estimate. The first requirement is related to the state equation while the second is related to the measured output. Therefore, a trade-off similar to the classical trade-off handled in the Kalman-filter framework by the state/measurement weighting can be adopted here using the following cost function in the decision variable $x^{(-)}$:

$$\hat{x}^{(-)} = \underset{\xi \in \mathbb{R}^n}{\operatorname{argmin}} \|G_1 \cdot \xi - S_1\|_{Q_1}^2 + \|G_2 \cdot \xi - S_2\|_{Q_2}^2$$

where Q_1 and Q_2 are the weighting matrices and hereafter the notation $\|x\|_Q^2 = x^T Q x$ is used. These matrices can be taken to be the inverse of the noise covariance matrices exactly as in the EKF framework, and the other matrices are defined by:

$$\begin{aligned} G_1 &= \Omega(\bar{U}_k) \\ G_2 &= \Phi(\bar{U}_k) \\ S_1 &= \bar{Y}_k - \Gamma(\bar{U}_k) \cdot \bar{U}_k - \bar{E}_k \\ S_2 &= A(u(k-1), w(k-1))\hat{x}(k-1) + B[u(k-1) \ w(k-1)]^T - \Psi(\bar{U}_k) \cdot \bar{U}_k \end{aligned}$$

With the above notations, an updating law $\hat{x}(k)$ for the state estimation can be defined as follows:

$$\hat{x}(k) = \Phi(\bar{U}_k) \cdot [(G_1^T \cdot Q_1 \cdot G_1 + G_2^T \cdot Q_2 \cdot G_2)^{\dagger} (S_1^T \cdot Q_1 \cdot G_1 + S_2^T \cdot Q_2 \cdot G_2)^T] \hat{x}_{k-1} + \Psi(\bar{U}_k) \cdot \bar{U}_k \quad (4.11)$$

leading to the following implicit form for the observer:

$$\hat{x}(k) = Obs(\hat{x}(k-1), \bar{Y}_k, \bar{U}_k)$$

Note that the observer equation (4.11) can be easily adapted in order to take into account the model (4.6), if \bar{Y}_k is used instead of \bar{U}_k in the expression of the matrices $\Phi(\bar{Y}_k)$, $\Omega(\bar{Y}_k)$, $\Gamma(\bar{Y}_k)$ and $\Psi(\bar{Y}_k)$.

4.3.3 Simulation Results

Some simulation results are presented in figures 4.8 and 4.9 in order to show the evolution of the norm of the state estimation error $\|x - \hat{x}\|^2$ of the MHO and the disturbance rejection of the offset ε given by (4.8) for the model (4.7). The weighting matrices Q_1 and Q_2 were set to the identities and the integrator gain $k_i = [3.5; 3.5]$ has been used in (4.8). The offset error $\varepsilon = [-3, -15]$ was introduced at $t = 0s$ and $\varepsilon = [-10, -5]$ at $t = 100s$, as shown in figure 4.9. Note that despite the presence of an offset error on the outputs, the asymptotic convergence of the estimation error is obtained.

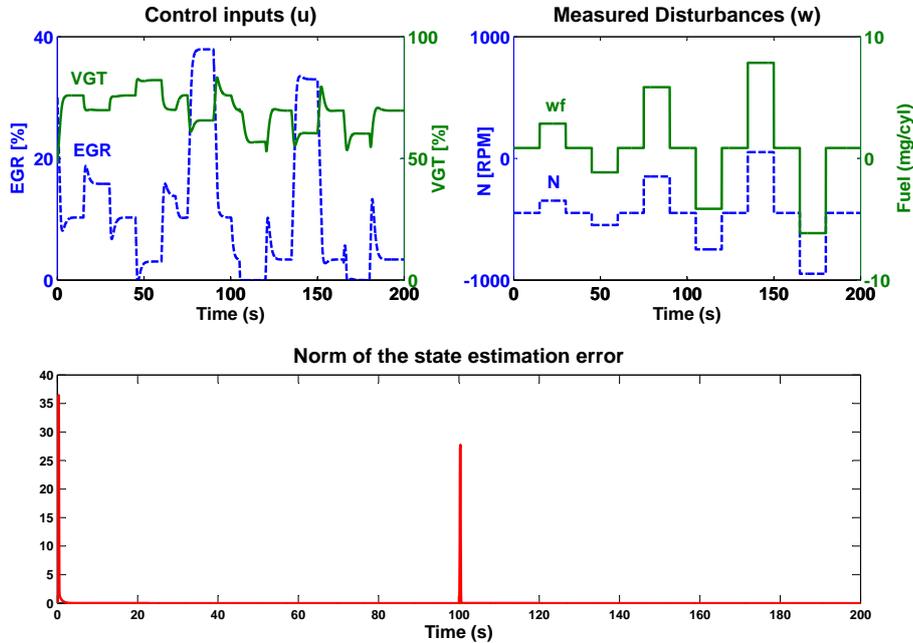


Figure 4.8: Simulation results related to the state estimation error. An offset error ε is inserted at $t = 0s$ and $t = 100s$ and the observer is still able to recover the state.

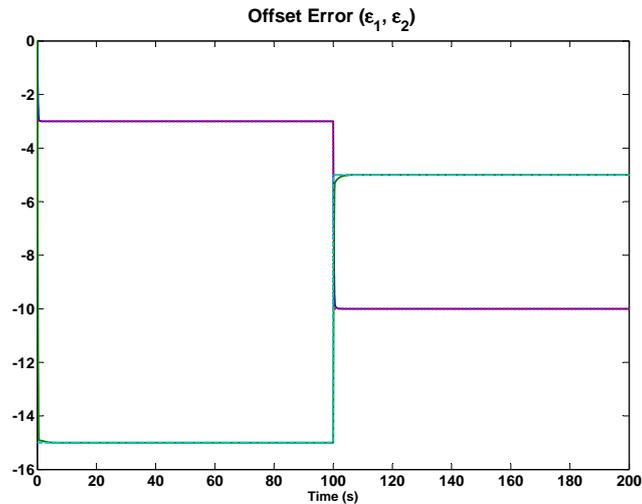


Figure 4.9: Simulation results showing the evolution of the offset error estimation. The disturbance ϵ inserted at $t = 0s$ and $t = 100s$ is quickly determined by the observation scheme.

Once the observation problem is solved, the next step is to design the output feedback. This is done using a parameterized NMPC strategy following the lines developed in chapter 3. This design of the parameterized NMPC is described in the following section.

4.4 Control Design

4.4.1 Introduction

In this section, a parameterized NMPC scheme is proposed for the diesel engine air path control problem. First, some important model features are underlined. In particular, it is shown that the computation of the stationary steady regime is quite crucial for the NMPC design. Afterwards, a parameterized NMPC scheme is proposed that involves the state observer developed in the previous section. Finally, it is underlined that the proposed parameterized NMPC can be used with more elaborated and complex models of the diesel engine.

4.4.2 Some Considerations About the Engine's Behavior

In addition to the model nonlinearities, the engine's behavior has two more important characteristics that must be taken into account, namely, the non-minimum phase and the open-loop behavior. These features are investigated hereafter.

4.4.2.1 Non-minimum Phase

The diesel engine is a highly coupled system. To show this, assume that one opens the EGR valve. This means that more exhaust gas is recirculated into the intake manifold resulting in a slight increase in MAP. On the other hand, this action implies that less exhaust gas is going to the turbine, which slows down the turbocharger, and hence decreasing MAP. Such situations are very common in diesel engines, introducing a non-minimum phase behavior.

Basically, the non-minimum phase behavior can be observed on the output evolution as shown in figure 4.10. Namely, the outputs are forced to go to opposite direction before going to the right one. For control design, this is explained by the presence of instable zeros in the system. This generally makes the task of the controller rather difficult.

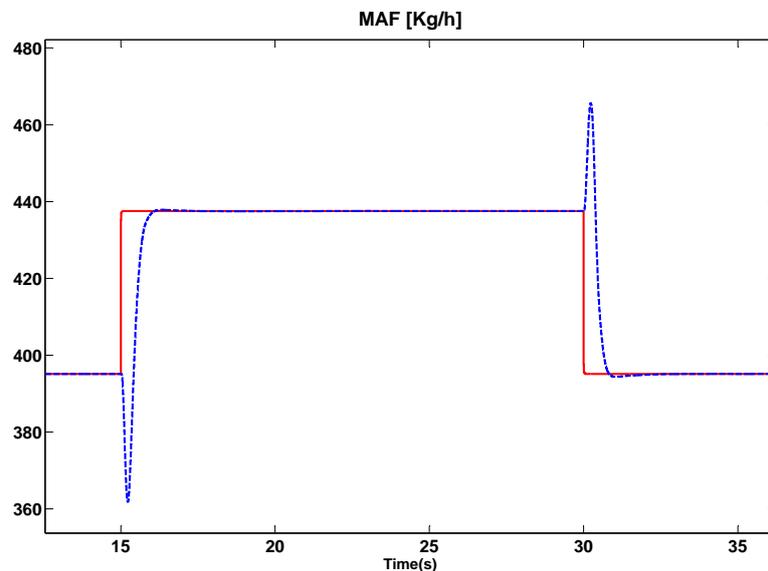


Figure 4.10: Typical behavior of the mass air flow showing a non-minimum phase property.

The implication of this property on the NMPC design is that the prediction horizon must be long enough to overcome this opposite response transient. This requirement makes the real-time issue even harder since too short prediction horizons cannot be considered and longer prediction horizons are necessarily more demanding in terms of on-line computation.

4.4.2.2 Open-Loop Behavior

A second important point to emphasize concerns the open-loop behavior of the diesel engine model. In fact, the system is open-loop stable. Figure 4.11 illustrates the outputs when a step sequence is used as input. This means that for a given value of EGR, VGT, speed and fuel injection, the outputs MAF and MAP asymptotically converge to some steady value that depend on the input values.

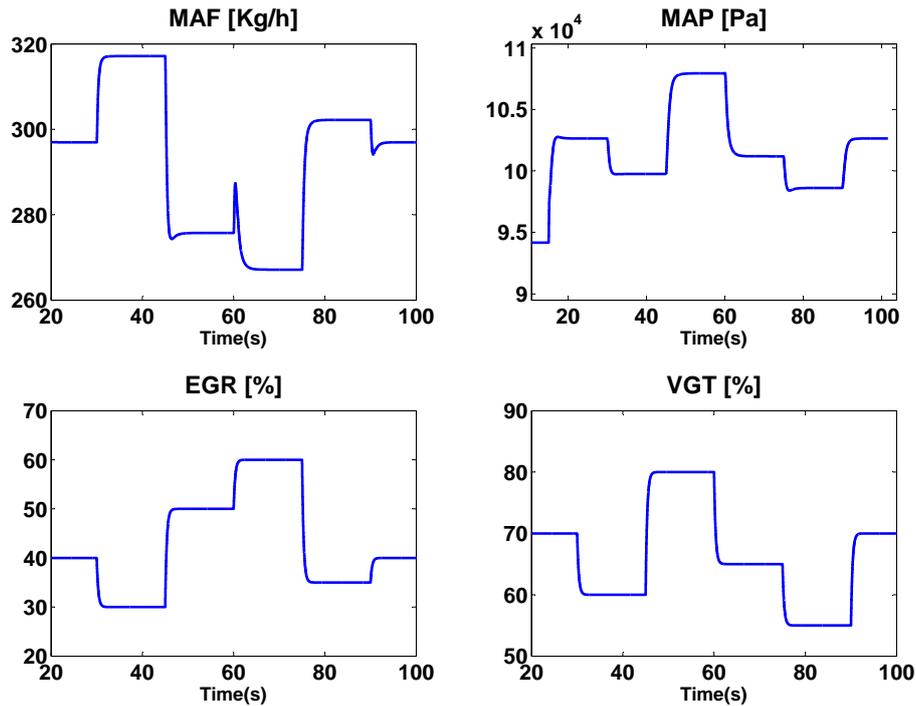


Figure 4.11: Open-loop behavior at $N_e=1800$ RPM and $w_f=20$ mg/cyl under a step sequence of EGR and VGT. This simulation was performed with the full nonlinear model (4.3).

The open-loop stability is an important advantage for the control design, especially for the parameterized NMPC scheme. Consequently, the next step consists in finding

the correct steady solution for a given set-point of MAP and MAF. This is shown in the next section.

4.4.3 The Steady Solution

Recall that NMPC is based on the prediction of the state evolution under some control profile. The latter depends on a finite number of parameters to be determined by on-line optimization. When the system model is open-loop stable, it is interesting to use the static gain map directly in the parametrization of the control profile in order to guarantee an asymptotically vanishing tracking error. By doing so, the degrees of freedom of the parametrization are used to improve the transient behavior of the tracking error. Consequently, computing the nonlinear static gain between the input and the regulated output is the first step in the definition of the control parametrization. In the sequel, the static gains of the two models described above, namely (4.6) and (4.7), are successively computed.

4.4.3.1 Static Gain of the Model (4.6)

As mentioned previously, the set-points y^d are generated by another control loop that manages the emission level. Assume that a set-point y^d is given. In order to compute the corresponding steady state x_d and steady control u_d for a given value of the measured disturbance vector w , one has to solve the following equations:

$$x_d = A(y^d)x_d + B_1u_d + B_2w \quad (4.12)$$

$$y^d = C(y^d)x_d + \varepsilon \quad (4.13)$$

However, it is needless to say that in a constrained environment (in the presence of saturations on the actuators for instance), there are limitations on the achievable values y^d for which equations (4.12)-(4.13) admit admissible solutions. One must invoke the set of achievable set-points that is necessarily bounded.

Then, the main idea is to compute the maximum μ_d^* over all values of $\mu_d \in [0, 1]$ such that the modified set-point $y^* = \mu_d^* \cdot y^d$ is achievable, as can be seen in figure 4.12. This may lead to the following equations in which μ_d has to be added to the unknown vector to yield the new unknown vector (x_d, u_d, μ_d) that characterizes the achievable steady state:

$$x_d = A(\mu_d \cdot y^d)x_d + B_1u_d + B_2w \quad (4.14)$$

$$\mu_d \cdot y^d = C(\mu_d \cdot y^d)x_d + \varepsilon \quad (4.15)$$

However, solving the above optimization problem would be cumbersome since the unknown μ_d appear in the arguments of the nonlinear maps $A(\cdot)$ and $C(\cdot)$. That is why the above optimization problem is replaced by the following one by still using y^d in the matrices $A(\cdot)$ and $C(\cdot)$ and using μ_d in the l.h.s of (4.15). This amounts to consider the following equations:

$$x_d = A(y^d)x_d + B_1u_d + B_2w \quad (4.16)$$

$$\mu_d \cdot y^d = C(y^d)x_d + \varepsilon \quad (4.17)$$

which is clearly an approximation of the original task but that leads to a tractable Linear Programming (LP) problem as it is stated in the following definition:

Definition 4.1. *Given a desired steady state y^d and a measured disturbance level w , the modified set-point is defined by $y^* = \mu^* \cdot y^d$ where μ^* is obtained by solving in the unknown vector*

$$(x_d^T \quad u_d^T \quad \mu_d)^T \in \mathbb{R}^{n+m+1}$$

the following Linear Programming problem:

$$\begin{aligned} LP(y^d, w) : \quad & \max \quad \mu_d \quad \text{under} \\ & \begin{pmatrix} \mathbb{I}_n - A(y^d) & -B_1 & 0_{nx1} \\ C(y^d) & 0 & -y^d \end{pmatrix} \cdot \begin{pmatrix} x_d \\ u_d \\ \mu_d \end{pmatrix} = \begin{pmatrix} B_2 \cdot w \\ \varepsilon \end{pmatrix} \\ & u_d \in [u_{min}, u_{max}] \\ & \mu_d \in [0, 1] \end{aligned}$$

The solution of the above LP is denoted by:

$$(x^*, u^*, \mu^*) \quad (4.18)$$

A schematic view of the set of achievable set-points y^d under $w = 0$ (those for which one obtains $\mu^* = 1$ when solving $LP(y^d, 0)$) is shown in figure 4.12.

The green region of the figure 4.12 shows the achievable zones, those for which the solution of the $LP(y^d, 0)$ gives $\mu^* = 1$. No controller can achieve a pair that is outside this region. In the red region the set point must be modified to be the closest one ($0 \leq \mu^* < 1$).

4.4.3.2 Static Gain of the Model (4.7)

Contrary to the preceding model, the feasible outputs can not be directly determined for the model (4.7) since the model does not depend on y anymore. Instead

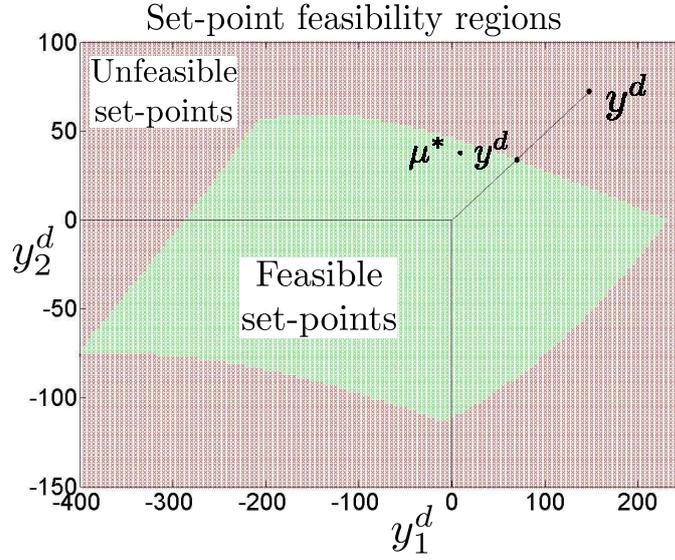


Figure 4.12: Schematic view of the feasible set-points, at $w = (0, 0)$.

of solving a simple linear program to obtain the steady control, in this case, a simple optimization problem must be solved. More precisely, given the measured disturbance vector w and the desired value y^d , the steady control is computed by solving the following two-dimensional optimization problem:

$$\begin{aligned}
 u^*(w, y^d) &:= \arg \min_{u_d \in [u_{min}, u_{max}]} \|y_c(u_d, w) - y^d\|^2 \\
 y_c(u_d, w) &= C(u_d, w)[\mathbb{I}_n - A(u_d, w)]^{-1} \cdot [B \cdot u_d + G \cdot w] + \varepsilon
 \end{aligned} \tag{4.19}$$

Moreover, having the steady control u^* , the steady state x^* can be directly obtained according to:

$$x^*(u^*, w) = [\mathbb{I}_n - A(u^*, w)]^{-1} \cdot [B \cdot u^* + G \cdot w] \tag{4.20}$$

Figure 4.13 illustrates the computation of u^* , by proceeding 10 iterations. The optimization routine that is used to solve (4.19) is presented in section 4.5.

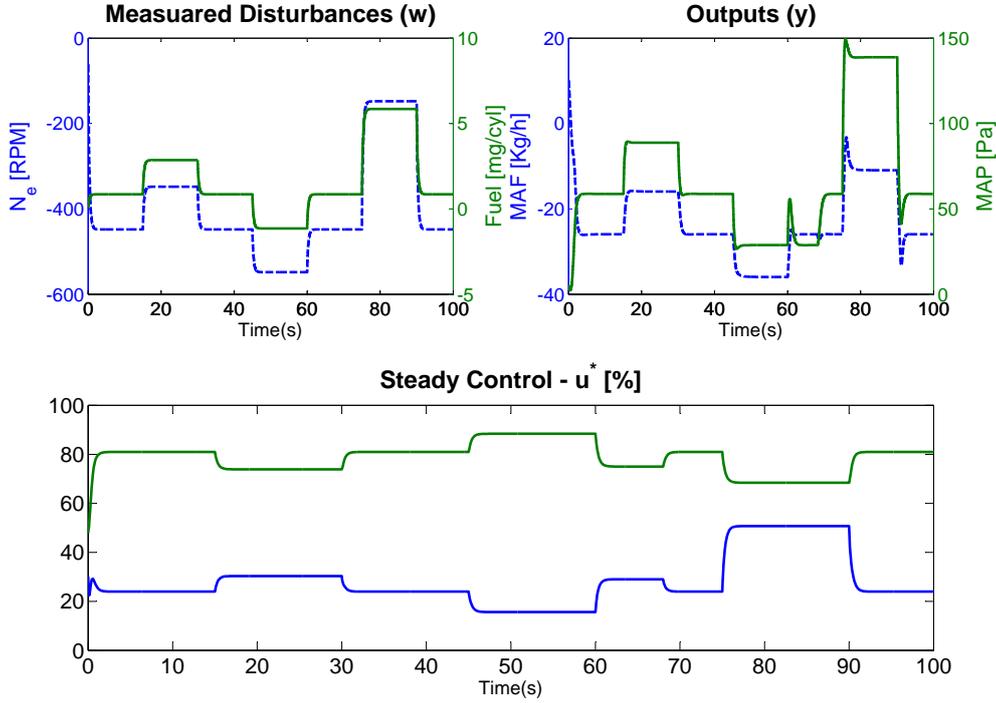


Figure 4.13: Evolution of u^* as a solution of the optimization problem (4.19), with 10 iterations, according to the variations of w and y .

It is worth mentioning that the particular structure of (4.7) allows to derive an analytical solution for u^* , and hence for x^* . But, it is worth noting that, the present formulation represents a general way to find the stationary control which is a crucial point to keep a generic definition of the control structure that may be used for general nonlinear models of the system under study. In the next sections, it is shown how this steady control is incorporated in the NMPC formulation in order to derive a real-time compatible output feedback scheme. According to table 4.1, section 4.4.4 presents the parameterized approach for models (4.6) and (4.7) while section 4.4.5 for the model (4.3)-(4.5) where a general framework for diesel engines is presented.

4.4.4 Parametric NMPC Formulation for Diesel Engines

The parametrization of the control sequence involves the stationary control u^* together with a simple temporal parametrization of the future control sequence. In order to do this, at the current instant k , the controller needs the state $x(k)$ or its estimated value $\hat{x}(k)$ provided by the observer according to (4.11), the actual value of the control input $u(k)$, already scheduled according to the computation performed over the sampling period $[k-1, k]$ and the values of the measured disturbances $w(k)$.

While in standard NMPC formulations, all the components of the control sequence \mathbf{u} are taken as degrees of freedom, here the following parametrization is used in order to get a decision variable of low dimension which is decoupled from the choice of the prediction horizon:

$$\mathbf{u}(i\tau_s + t) = \text{Sat}_{u_{min}}^{u_{max}} \left(u^* + \alpha_1 \cdot e^{-\lambda \cdot i \cdot \tau_s} + \alpha_2 \cdot e^{-q \cdot \lambda \cdot i \cdot \tau_s} \right) \quad \text{for } t \in [(k-1)\tau_s, k\tau_s[\quad (4.21)$$

where $i \in \{0, \dots, N-1\}$, τ_s is the sampling period, $\lambda > 0$, $q \in \mathbb{N}$ are tuning parameters, $\alpha_1, \alpha_2 \in \mathbb{R}^m$ are the coefficients to be determined and they are more explained below and Sat is a saturation map $\text{Sat} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ defined as follows:

$$\text{Sat}_{u_{min}}^{u_{max}}(u_i) = \begin{cases} u_{min}^i & \text{if } u_i \leq u_{min}^i \\ u_{max}^i & \text{if } u_i \geq u_{max}^i \\ u_i & \text{otherwise} \end{cases}, \quad i \in \{1, 2\} \quad (4.22)$$

Note that by using the expressions (4.21)-(4.22), the saturation constraints on the control input are structurally respected. As a matter of fact, at instant k , the continuity of the control sequence must be guaranteed. Therefore, using $i = 0$ in (4.21) leads to the following constraints:

$$u^* + \alpha_1 + \alpha_2 = u(k-1) \quad (4.23)$$

Moreover, the constraints on the variation rates of the control (4.2) can be respected provided that one meets the following constraints:

$$\alpha_1 \cdot (e^{-\lambda \cdot \tau_s} - 1) + \alpha_2 \cdot (e^{-q \cdot \lambda \cdot \tau_s} - 1) = p \cdot \delta_{max} \quad ; \quad p \in [-1, +1]^2 \quad (4.24)$$

where $p \in [-1, 1]^2$ is parameter vector. Note that the above expression simply states that the difference between two successive controls does not exceed a fraction $p \in [-1, 1]^2$ of the maximal allowable values δ_{max} . Therefore, one can put together the equations (4.23)-(4.24) to obtain a system of 4 equations in the four unknown $\alpha_i^{u_j}$ for $i = 1, 2$ and $j = 1, 2$ that become now dependent on the parameter vector $p \in [-1, +1]^2$ according to:

$$\begin{pmatrix} \alpha_1^{u_1}(p) \\ \alpha_2^{u_1}(p) \\ \alpha_1^{u_2}(p) \\ \alpha_2^{u_2}(p) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 & 0 & 0 \\ 0 & 0 & e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 \end{pmatrix}^{-1} \begin{pmatrix} u_1(k-1) - u_1^* \\ u_2(k-1) - u_2^* \\ p_1 \delta_{max}^1 \\ p_2 \delta_{max}^2 \end{pmatrix}$$

where the notation $\alpha_1 = [\alpha_1^{u_1}; \alpha_1^{u_2}]$ and $\alpha_2 = [\alpha_2^{u_1}; \alpha_2^{u_2}]$ is used. Injecting this in the equation of the control sequence leads to:

$$\mathbf{u}(i\tau_s + t, p) = \text{Sat}_{u_{min}}^{u_{max}} \left(u^* + \alpha_1(p) \cdot e^{-\lambda \cdot i \cdot \tau_s} + \alpha_2(p) \cdot e^{-q \cdot \lambda \cdot i \cdot \tau_s} \right) \quad (4.25)$$

The expression (4.25) clearly shows how the control profile depends on the parameter vector p introduced in (4.24).

As for the cost function, it is reasonable to define a stage cost that penalizes the error $y - y^d$ since the problem to be solved is the tracking of MAF and MAP. Therefore, given the desired set-point y^d , the best set of parameters to be used in the definition of optimal control sequence is provided by solving, during the sampling period $[k\tau_s, (k+1)\tau_s[$, the following optimization problem:

$$\hat{p} := \arg \min_{p \in \mathbb{P}} \left[\rho_x \cdot \|X(N, \hat{x}(k), p) - x^*(y^d, w)\| + \sum_{i=0}^{N-1} \|Y(i, \hat{x}(k), p) - Y_f^*(i, y^d, w)\|_{Q_y}^2 \right] \quad (4.26)$$

where $\hat{x}(k)$ is estimated state at instant k , $Y(i, \hat{x}(k), p)$ and $X(i, \hat{x}(k), p)$ are respectively, the output and state at instant $k+i$ based on the model being used under the open loop control profile defined by p over $[k, k+N-1]$ and starting from $\hat{x}(k)$. The state $x^*(y^d, w)$ is the stationary state computed above according to (4.18) or (4.20), $\rho_x > 0$ is some weighting coefficient used to enforce the constraint on the final state (for stability purposes), $Y_f^*(i, y^d, w)$ is the filtered version of the set-point used to avoid overshoot in the system response, namely:

$$Y_f^*(i, y^d, w) = y^*(y^d) + e^{-3\tau_s \cdot i/t_r} \cdot [y(k) - y^*(y^d)] \quad (4.27)$$

where t_r is the desired response time of the closed-loop system, and Q_y the square matrix of dimension m used to differently weight the outputs y_1 and y_2 , and has the following structure

$$Q_y = \begin{pmatrix} \frac{\rho_1}{\bar{y}_1} & 0 \\ 0 & \frac{\rho_2}{\bar{y}_2} \end{pmatrix}_{m \times m} \quad (4.28)$$

where ρ_1 and ρ_2 are the weighting terms of y_1 and y_2 respectively and \bar{y}_1 and \bar{y}_2 the normalization terms of each output.

Figure 4.14 summarizes the whole output feedback strategy, with the MHO representation and the parameterized NMPC. The data \bar{Y}_k is collected during a moving and fixed-size window $[(k-N_o+1)\tau_s, k\tau_s]$ together with the past measured inputs \bar{U}_k , both are used to estimate the past state $\hat{x}^{(-)}$ at instant $k-N_o+1$ and hence, the present state $\hat{x}(k)$ (4.11). This value is used in the definition of the NMPC strategy based on open-loop simulations of the saturated exponential p -dependent parametrization profile.

Note that more degrees of freedom can be used in (4.25). The general expression may be given by:

Note that in the above formulation, the constraints on the parameters $p_1, p_2 \in [-1, +1]^2$ are still valid, but the vector with the new extra parameters represented by $(p_3 \ p_4 \ \dots \ p_{2d_f+2})$ does not have any condition to impose structural constraints, as for the input derivatives, and they can be freely chosen. However, increasing d_f in order to introduce more dynamics in the system may not be appropriate. This is because the gain in the closed-loop quality may not be sufficient to justify the resulting increase of computational complexity.

4.4.5 A Black-Box NMPC Solution for Diesel Engines

The last section presented the parameterized approach using the exponential terms to build the control sequence \mathbf{u} . However, the solution is dependent on the computation of the steady control u^* . This is particularly due to the structure of the model being used, especially for the model (4.6), where the computation of u^* was practically direct, and almost no optimization procedure had to be done.

On the other hand, for the model (4.7), the way to compute the steady control is the first step to obtain a general NMPC procedure for diesel engines. This is because in (4.19), the u^* is obtained by solving an optimization problem. But in that specific case, the affine structure of the model has been used, and a kind of gray-box solution was obtained. Here, the aim is to define a formulation that does not rely on any particular mathematical structure of the model. Considering a general non linear system represented by:

$$\begin{aligned} x^+ &= f(x, u, w) \\ y &= g(x, u, w) \end{aligned}$$

where $f : \mathbb{R}^n \times \mathbb{R}^{2m} \rightarrow \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^{2m} \rightarrow \mathbb{R}^m$. In fact, starting from the structure of the control profile in (4.25), one can consider that u^* used in the expression of the control profile is no more obtained by external computation but is a part of the decision variable. Therefore, the new control sequence can be rewritten as follows:

$$\mathbf{u}(i\tau_s + t, p) = \text{Sat}_{u_{min}}^{u_{max}} \left([p_3 \ p_4]^T + \alpha_1(p) \cdot e^{-\lambda \cdot i \cdot \tau_s} + \alpha_2(p) \cdot e^{-q \cdot \lambda \cdot i \cdot \tau_s} \right) \quad (4.30)$$

Note that the new set of parameters in (4.30) is defined such as $p \in \mathbb{R}^4$ since the derivative terms remain the same. Then, the resulting parameter vector p must respect the following set of constraints.

$$\begin{aligned} (p_1 \ p_2)^T &\in [-1, +1]^2 \\ (p_3 \ p_4)^T &\in \mathbb{R}^2 \end{aligned}$$

The above expression shows that the parameters p_3 and p_4 can be freely chosen in \mathbb{R}^2 since the control sequence \mathbf{u} in (4.30) is already saturated between u_{min} and u_{max} . However, some constraints can be imposed in order to reduce the search space, for example, by introducing the constraints on the inputs (4.1), leading to the following condition:

$$(p_3 \ p_4)^T \in [u_{min}, u_{max}]^2 \quad (4.31)$$

In this case, p_3 and p_4 represent the stationary control u^* . Then, the computation of α_1 and α_2 can be done by solving the modified system of equations:

$$\begin{pmatrix} \alpha_1^{u_1}(p) \\ \alpha_2^{u_1}(p) \\ \alpha_1^{u_2}(p) \\ \alpha_2^{u_2}(p) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 & 0 & 0 \\ 0 & 0 & e^{-\lambda \cdot \tau_s} - 1 & e^{-q \cdot \lambda \cdot \tau_s} - 1 \end{pmatrix}^{-1} \begin{pmatrix} u_1(k-1) - p_3 \\ u_2(k-1) - p_4 \\ p_1 \delta_{max}^1 \\ p_2 \delta_{max}^2 \end{pmatrix}$$

However, this new formulation requires some few considerations about the cost function defined in (4.26).

It is worth noting that the formulation of the cost function presented in the previous section (4.26) is valid only for the empirical models (4.6) and (4.7). This is due to the fact that the weighing term on the terminal state depends on the computation of x^* which is provided by the exact value of u^* obtained in a previous step from a dedicated steady state calculation. As far as the full nonlinear model (4.3)-(4.5) is concerned, the stationary state x^* can not be obtained after u^* , since there is no more specific routine for the computation of the steady control. As a result, the terminal cost on the state in (4.26) must be changed. Therefore, for a general class of nonlinear models for diesel engines, the new cost function to be considered becomes:

$$\hat{p} := arg \min_{p \in \mathbb{P}} \left[\rho_x \cdot \|x_f(p) - f(x_f(p), u^{(N)}(p(\cdot)), w)\|^2 + \sum_{i=0}^{N-1} \|Y(i, \hat{x}(k), p) - Y_f^*(i, y^d, w)\|_{Q_y}^2 \right] \quad (4.32)$$

where $x_f(p) = X(N, \hat{x}(k), p)$. Note that the terminal cost is now defined as the norm of the difference of two successive values of the state, at the end of prediction horizon, under the control profile depending on p . In fact, this is the classical way to address stability in predictive control formulations. However, first simulation results (section 4.6.3) showed that the terminal cost can be dropped by setting ρ_x to 0 since in this particular case the system is open-loop stable.

Therefore, with the above modifications, the computation of p does not rely on any specific model structure, making the present solution appropriate to be used as a

black-box NMPC controller for diesel engines or for any other non linear system which has similar behavior with this model.

To summarize, a general framework for diesel engines can be formalized as follows:

- **General nonlinear system**

$$\begin{aligned}x^+ &= f(x, u, w) \\ y &= g(x, u, w)\end{aligned}$$

- **Control Law**

$$\mathbf{u}(t) = \text{Sat}_{u_{min}}^{u_{max}} \left([p_3 \ p_4]^T + \alpha_1(p)e^{-\lambda t} + \alpha_2(p)e^{-q\lambda t} \right) \ ; \ \alpha_i \in \mathbb{R}^2$$

- **Cost Function**

$$\hat{p} := \arg \min_{p \in \mathbb{P}} \left[\rho_x \cdot \|x_f(p) - f(x_f, u^{(N)}(p(\cdot)), w)\|^2 + \sum_{i=0}^{N-1} \|Y(i, \hat{x}(k), p) - Y_f^*(i, y^d, w)\|_{Q_y}^2 \right]$$

with $x_f(p) = X(N, \hat{x}(k), p)$

- **Constraints**

$$\begin{aligned}(p_1 \ p_2)^T &\in [-1, +1]^2 \\ (p_3 \ p_4)^T &\in \mathbb{R}^2 \text{ or } (p_3 \ p_4)^T \in [u_{min}, u_{max}]^2\end{aligned}$$

4.5 Optimization Process

4.5.1 Low Dimensional Optimization

Regardless the choice of the optimization subroutine, the complexity of the optimization task heavily depends on the problem formulation, in particular, the control parametrization. Some simulations are presented in the sequel in order to better visualize the resulting optimization problem and the evolution of the parameter vector p . For instance, let us define the following scenario :

- The aim is to track the desired reference $y^d = [60 \text{ Kg/h}; -25 \text{ hPa}]$.
- The model to be used is the one depending on the inputs (4.7). Therefore, the parameter vector is $p \in \mathbb{R}^2$ (since the computation of the steady control is done by (4.19)) and the cost function is represented by (4.26).
- The initial state of the observer has been set to 0.
- The prediction horizon has been set to $N = 30$ and the tuning parameters λ , q to 1 and 5 respectively and the weighting term ρ_x on the state to 0.001. It is worth noting that the sampling period τ_s is 50 ms since the model is (4.7).
- This simulation has been performed at $w = [448 \text{ RPM}; 0.8 \text{ mg/st}]$.
- Three particular instants t_i , $i \in \{1, 2, 3\}$ have been chosen to illustrate the shape of the performance index $J(\cdot, p, \hat{x}(t_i))$.
- The condition to stop the optimization routine (which is explained in the next topic) is the number of function evaluations n_{fe} and in this case it has been set to 10.

Figures 4.15 and 4.16 show the evolution of the parameters p_1 and p_2 as well as the system outputs. Note that, according to the definition of the parametrization, the constraint on the rate of change of the control input is active when p is saturated (at -1 or 1).

The shape of the cost function J gives a relevant indication on the complexity of the optimization problem. This can be seen in figure 4.17, which clearly shows how the low dimensional control parametrization adopted above leads to a well posed optimization problem (the cost function seems to be convex). In the next topic, some details are given concerning the optimization algorithm that has been used in simulations and also to validate experimentally the proposed parameterized NMPC formulation.

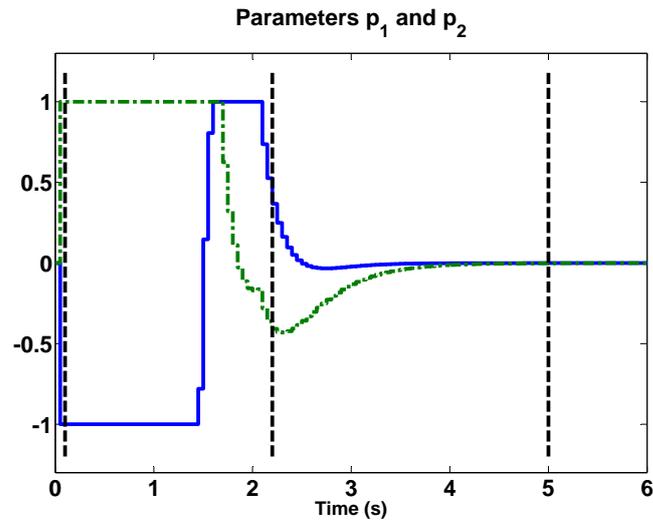


Figure 4.15: Simulation results showing the evolution of p_1 et p_2 to track $y_d = [60 \text{ Kg/h}; -25 \text{ hPa}]$ (Fig. 4.16), with $n_{fe} = 10$. Three particular instants ($t_1 = 0.1s$, $t_2 = 2.2s$ and $t_3 = 5s$) have been used to illustrate the shape of cost function (figure 4.17). Note that when p is saturated at 1 or -1 the maximum derivatives δ_{max} are allowed according to (4.24).

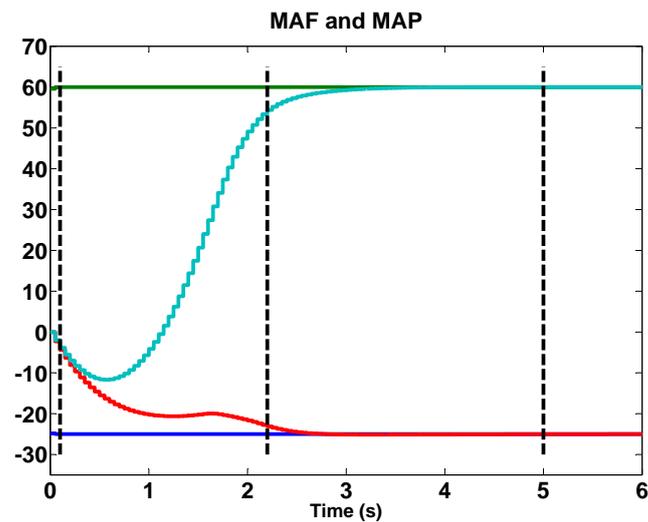


Figure 4.16: Simulation results showing the measured values of MAF and MAP (deviations from the central values) and desired outputs y_d . Vertical lines define the three instants used to illustrate the shape of the cost function (Figure 4.17).

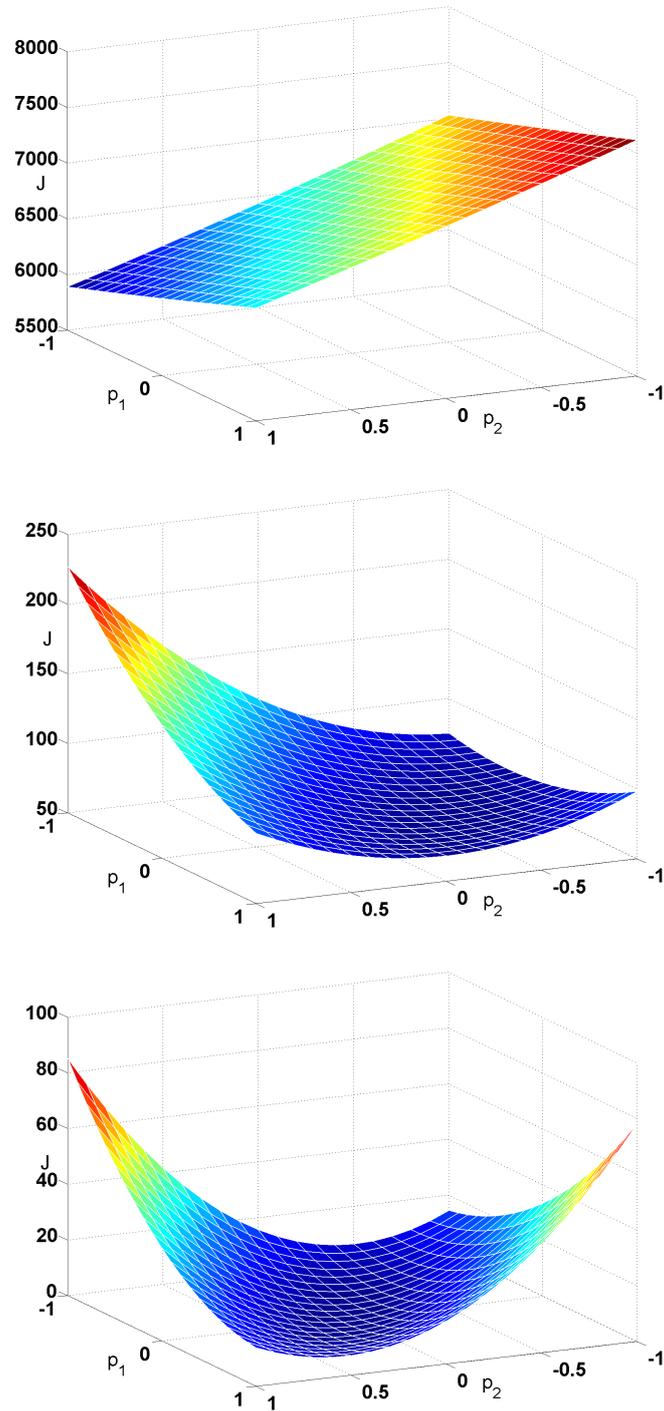


Figure 4.17: The shape of the cost function $J(\cdot, p, \hat{x}(t_i))$ for the three particular instants depicted on Figures 4.15 and 4.16 ($t_1 = 0.1$ s (top), $t_2 = 2.2$ s (middle), $t_3 = 5.0$ s (bottom)), with $n_{fe} = 10$. Note the decrease of the cost function and the efficiency of the parameterized approach. This efficiency relies on the convex shape and the low dimensional underlying optimization problem.

4.5.2 A New Mixed SQP/Gradient/Trust Region Method

First, it is worth mentioning that it is not in our mind to claim having any contribution in the domain of optimization algorithms. The well conditioned optimization problem mentioned above can be solved using any available software. Nevertheless, the fact that the optimization solver has to be embedded in the experimental facility incited us to write a self-contained code that does not depend on any sort of numerical libraries. The resulting optimization code that has then been used experimentally to validate the efficiency of the control law is presented in this section for completeness.

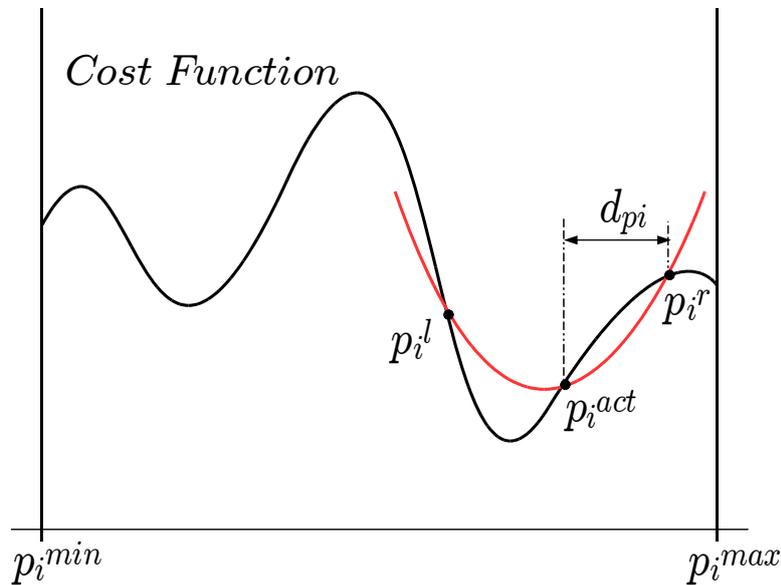


Figure 4.18: Schematic view of the preparation phase of the scalar SQP routine for the component i of the parameter vector p .

The algorithm is based on successive steps of scalar quadratic programming followed by an attempt to perform a gradient descent step. First, let us introduce the scalar SQP procedure. For a given value of p , the actual value of the cost function J_{act} can be evaluated. A first parameter $d_p \in \mathbb{R}^{n_p}$ is introduced in order to describe the trust region around p . In fact, d_p represents the displacement from p , in order to compute the cost function at the left and at the right to p . Then, the scalar procedure performs three function evaluations and with these values of the cost function, a quadratic approximation can be obtained as shown in figure 4.18. Note that the values of p must be saturated from p_{min} and p_{max}

The minimum of the so obtained scalar quadratic function is computed to define a candidate value p_i^{cand} for the updated component p_i . A candidate cost J_{cand} is then evaluated leading to two possibilities according to whether J_{cand} is lower or greater

than the current value J_{act} . If J_{cand} is lower, then the corresponding candidate parameter vector becomes the current one. Moreover, the value of d_{p_i} is increased since the quadratic approximation leads to a success in the direction i . The increase in d_{p_i} is obtained using a parameter $\beta^+ > 1$ according to

$$d_{p_i} \leftarrow \beta^+ \cdot d_{p_i} \quad (4.33)$$

Figure 4.19 illustrates this possibilities [$J_{cand} < J_{act}$].

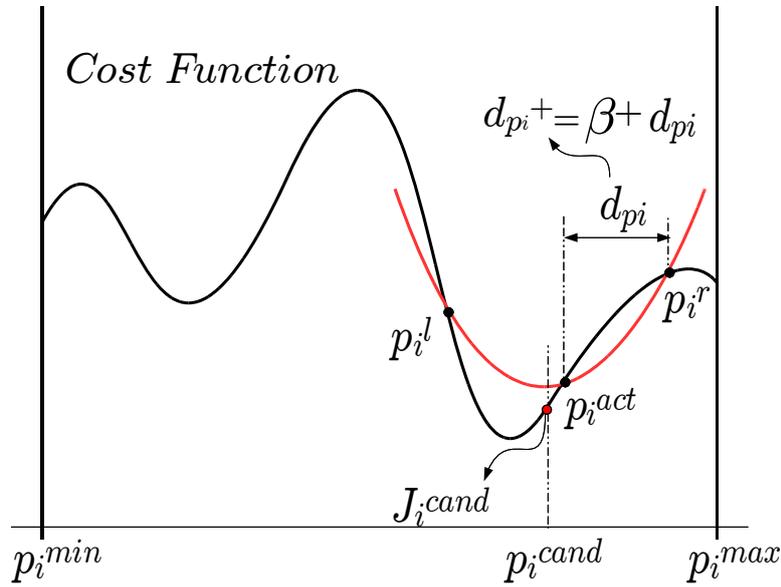


Figure 4.19: The SQP scalar algorithm performs successively scalar SQP's on each component i , each followed by an adaptation of the corresponding trust region. In this figure, the test is successful since the candidate cost function is lower than its current value. Therefore the i -th component of the step size d_{p_i} is using the successful updating parameter $\beta^+ > 0$.

If the second possibility holds, namely, if the test $J_{cand} < J_{act}$ fails, the trust region must be reduced since the quadratic approximation is bad. The search region is then reduced by updating d_{p_i} according to:

$$d_{p_i} \leftarrow \max \left\{ d_p^{min}, \beta^- \cdot d_{p_i} \right\} \quad (4.34)$$

where $\beta^- \in]0, 1[$. The saturation function is used in order to avoid the collapse of d_{p_i} that may prevent future success in the corresponding direction. This procedure is illustrated in figure 4.20

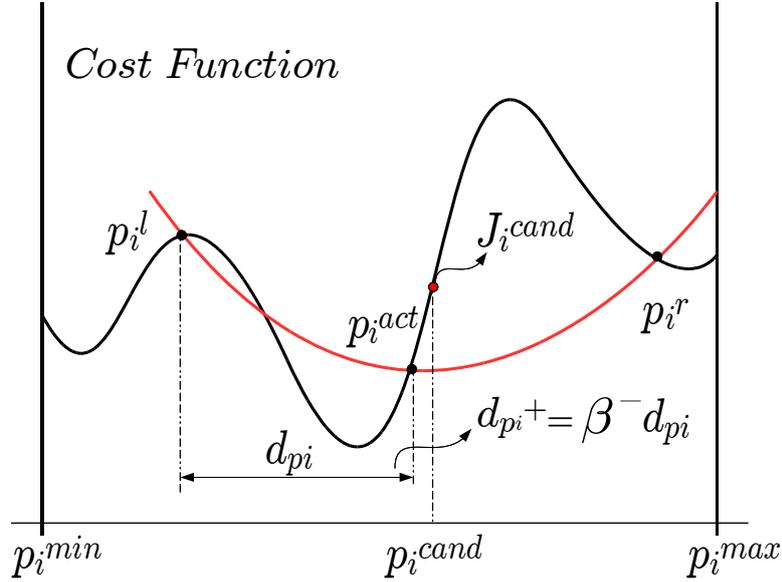


Figure 4.20: Situation when the test $J_{cand} < J_{act}$ fails. As a result, the trust region parameter d_{p_i} is decreased using the parameter β^- .

This procedure is performed in all the directions $i \in \{1, \dots, n_p\}$ and all the components of the step vector d_p are therefore, updated. By doing so, the vector d_p may contain a precious information concerning the descent directions of the cost function. This suggests that, after each complete cycle of scalar SQP's over the components, the resulting vector $d_p \in \mathbb{R}^{n_p}$ can be used to construct an approximation of the gradient along which potentially successful steps can be attempted. Here again, a trust region along the gradient is defined and updated according to the success or failure of such step.

The gradient step performs only one function evaluation along the direction given by the $\alpha_g \cdot d_p$ vector, where α_g is the step size associated to the gradient vector. As for the previous case, α_g is multiplied by $\beta^+ > 1$ if the gradient step is successful, according to:

$$\alpha_g \leftarrow \beta^+ \cdot \alpha_g \quad (4.35)$$

On the other hand, the fail parameter $\beta^- < 1$ is used if the test is not successful. As in the previous case, α_g is saturated at $\alpha_{g_{min}}$ to avoid excessively small values:

$$\alpha_g \leftarrow \max\{\alpha_g^{min}, \beta^- \cdot \alpha_g\} \quad (4.36)$$

This procedure is repeated as long as the cost function decreases (the gradient test is successful) and stops when the gradient test fails to undertake a new cycle of scalar

SQP/trust region iterations. If the maximal number of function evaluations n_{fe} is reached, the optimization problem ends and returns the current set of suboptimal parameters p_{sopt} . This results in the following algorithm:

Algorithm 1 Main Program : Compute a sub-optimal value p_{sopt}

```

Initialization:  $d_p \leftarrow d_p^0; \alpha_g \leftarrow \alpha_g^0; p \leftarrow p^0$ 
 $J \leftarrow \text{Cost Function}(p)$ 
 $iter \leftarrow 1$ 
while  $iter \leq n_{fe}$  do
  for  $i = 1$  to  $n_p$  do
     $[p_i, d_{p_i}, J, grad_i] \leftarrow \text{Scalar SQP Procedure}(p, d_{p_i}, J)$ 
     $iter \leftarrow iter + 3$ 
     $i \leftarrow i + 1$ 
  end for
   $Success_{Gradient} \leftarrow 1$ 
  while  $Success_{Gradient} = 1$  and  $iter \leq n_{fe}$  do
     $[p, Success_{Gradient}] \leftarrow \text{Gradient Procedure}(p, grad, \alpha_G, J)$ 
     $iter \leftarrow iter + 1$ 
  end while
end while
 $p_{sopt} \leftarrow p$ 
Return( $p_{sopt}$ )

```

Algorithm 2 Scalar SQP Procedure

```

 $J_{left} \leftarrow \text{Cost Function}(p - d_p)$ 
 $J_{right} \leftarrow \text{Cost Function}(p + d_p)$ 
 $[a, b, c] \leftarrow \text{Quadratic Polinomial}(J_{left}, J_{act}, J_{right}, p, d_p, p_{min}, p_{max})$ 
 $p_{candidate} \leftarrow \text{Sat}(\text{Min Value}(a, b, c), p_{min}, p_{max})$ 
 $J_{candidate} \leftarrow \text{Cost Function}(p_{candidate})$ 
if  $J_{candidate} < J_{act}$  then
   $p \leftarrow p_{candidate}$ 
   $J_{act} \leftarrow J_{candidate}$ 
   $d_p \leftarrow \beta^+ d_p$ 
   $grad \leftarrow d_p$ 
else
   $d_p \leftarrow \text{Max}(\beta^- d_p, d_{pmin})$ 
   $grad \leftarrow 0$ 
end if
Return( $p, d_p, J, grad$ )

```

Algorithm 3 Gradient Procedure

```

for  $i = 1$  to  $n_p$  do
   $p_{candidate} \leftarrow Sat(p_i - \alpha_g \cdot grad_i, p_{min}, p_{max})$ 
end for
 $J_{candidate} \leftarrow \text{Cost Function}(p_{candidate})$ 
if  $J_{candidate} < J_{act}$  then
   $p \leftarrow p_{candidate}$ 
   $J_{act} \leftarrow J_{candidate}$ 
   $\alpha_g \leftarrow \beta^+ \alpha_g$ 
else
   $\alpha_g \leftarrow \text{Max}(\beta^- \alpha_g, \alpha_{g_{min}})$ 
end if
Return( $p, \alpha_g, J$ )

```

The sub-programs used in these algorithms are described below:

- Cost Function(p): Returns the cost function value at p
 - Quadratic Polynomial($J_1, J_2, J_3, p, d_p, p_{min}, p_{max}$): Returns the parabola coefficients given the points ($Max(p-d_p, p_{min}), J_1$), (p, J_2) and ($Min(p+d_p, p_{max}), J_3$)
 - Min Value(a, b, c): Returns the minimum value $-b/2a$
 - Sat(a, min, max): Returns a or its saturated value min or max if $a < min$ or $a > max$ respectively.
 - Max(a, b): Returns the maximum value between a and b
 - Min(a, b): Returns the minimum value between a and b
-

4.6 Simulation Results

In this section, some simulation results are proposed in order to evaluate the performance of the proposed NMPC scheme and its real-time implementability. The programs related to the MHO and the NMPC were developed in C language and integrated to a MATLAB environment via S-functions and MEX-files. The hardware configuration used to perform computations was a Pentium 4, 3GHz. This section is split into three parts related to the three different nonlinear models presented previously showing the efficiency of the controller regardless the model being used. No extra degree of freedom was used in the simulations ($d_f=0$) and the parameters of the solver presented in the previous section are shown in table 4.2:

Table 4.2: Parameters of the solver used in simulations

Parameter	Value	Expression
d_p	0.1	(4.33)-(4.34)
α_g	0.1	(4.35)-(4.36)
$[\beta^-, \beta^+]$	[0.8,1.2]	(4.33)-(4.36)

4.6.1 Simulations Using the Model Depending on the Outputs

This model was the first nonlinear model identified by our colleagues from the University of Johannes Kepler, Linz. The results shown hereafter have been presented in [Murilo et al., 2009a]. All the simulations were performed with a sampling period of 10 ms since the model has been identified with this sampling period. The parameter values used in the simulations are shown in table 4.3. The weighting matrices Q_1 and Q_2 used in the the MHO equation (4.11) have been taken equal to the identity in their respective spaces and the initial estimated state \hat{x}_0 has been taken uniformly equal to 10 in order to check the convergence of the observer and the normalization terms \bar{y}_1 and \bar{y}_2 in (4.28) were set to 1.

Figure 4.21 shows a sequence of filtered steps that are used as reference values for the MAF and the MAP where no measured disturbances are considered ($w = 0$). Note that the constraint on the control input are respected. The bottom right subplot shows the evolution of the parameters p_1 and p_2 . Note that by the very definition of the parametrization, when p is saturated, the constraint on the rate of change of the control input is active. The bottom left subplot shows the corresponding computation time needed to perform the 10 allowed function evaluations for the NMPC controller as well as the computations related to the observer. It clearly

Table 4.3: Parameters used in simulations - Model (4.6)

Parameter	Value	Expression	Parameter	Value	Expression
N	30	(4.26)	N_o	30	(4.11)
λ	1	(4.25)	q	2	(4.25)
τ_s	0.01s	(4.6)	ρ_x	1000	(4.26)
ρ_1	1	(4.28)	ρ_2	1	(4.28)
t_r	1s	(4.27)	n_{fe}	10	-
$[u_{min}, u_{max}]^2$	$[0, 100]^2\%$	(4.1)	$[-\delta_{max}, \delta_{max}]^2$	$[-1, 1]^2\%/\tau_s$	(4.2)

indicates that the computation time never exceeds the sampling period of 10 *ms* thus assessing the real-time implementability of the proposed framework.

The second scenario shown by figure 4.22 represents a more realistic behavior for the set-point reference, which is a variable signal. Note that the controller is able to track it with a quite nice performance while the constraints on the control input are always respected.

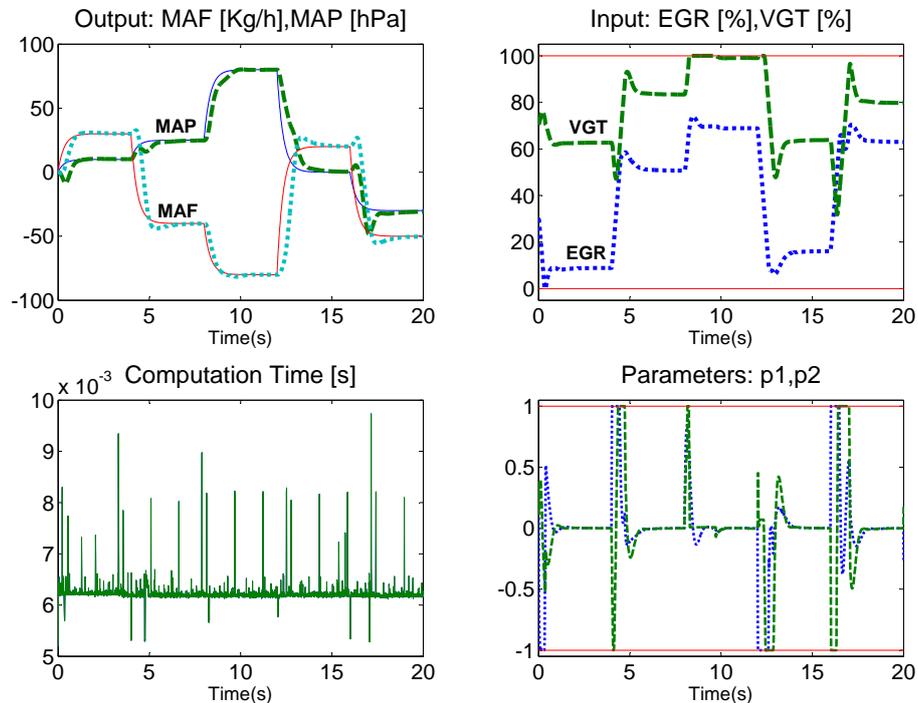


Figure 4.21: Evolution of the closed loop system with a sequence of filtered steps (Model (4.6)).

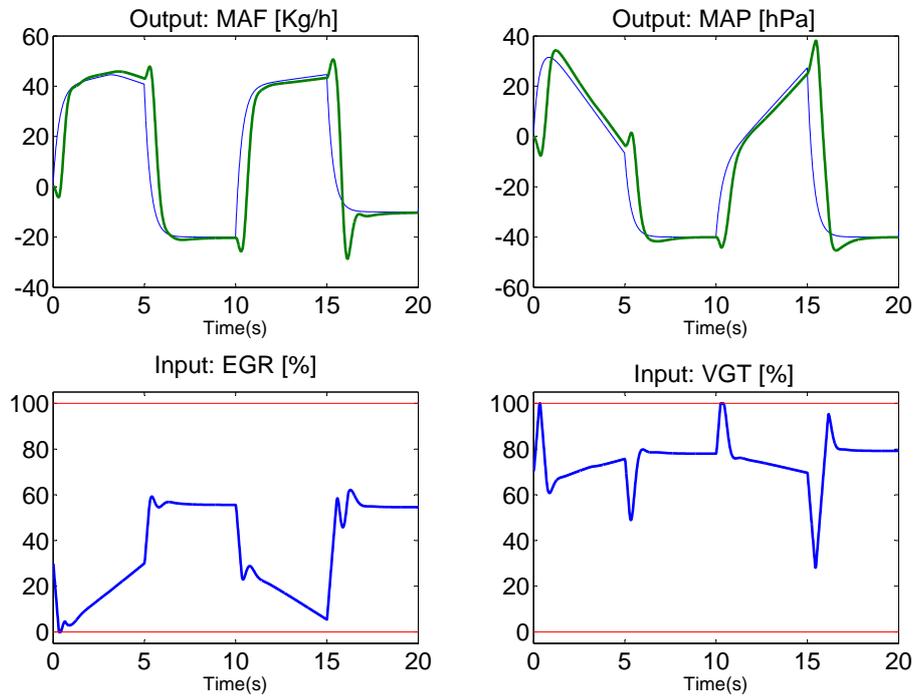


Figure 4.22: Evolution of the closed loop system with a variable signal reference (Model (4.6)).

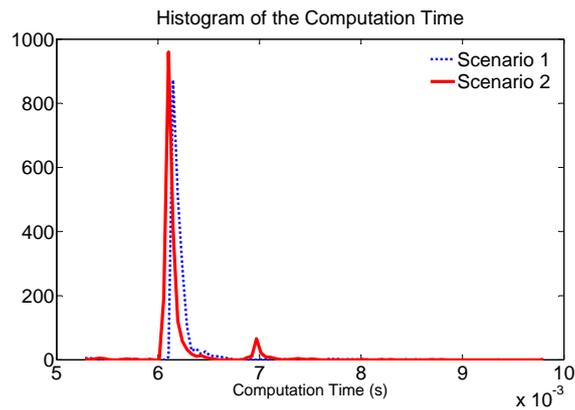


Figure 4.23: Histograms of the computation time needed at each sampling period to perform the observation task and the $n_{fe} = 10$ function evaluations of the optimization procedure (Model (4.6)).

Figure 4.23 shows histograms of the computation times for the preceding scenarios. It can be noticed that the computation time rarely exceeds 7 ms and never exceeds the allowable computation time of 10 ms . The last scenario (Fig. 4.24) shows the closed-loop behavior in response to step changes in the component of the measured disturbance vector w .

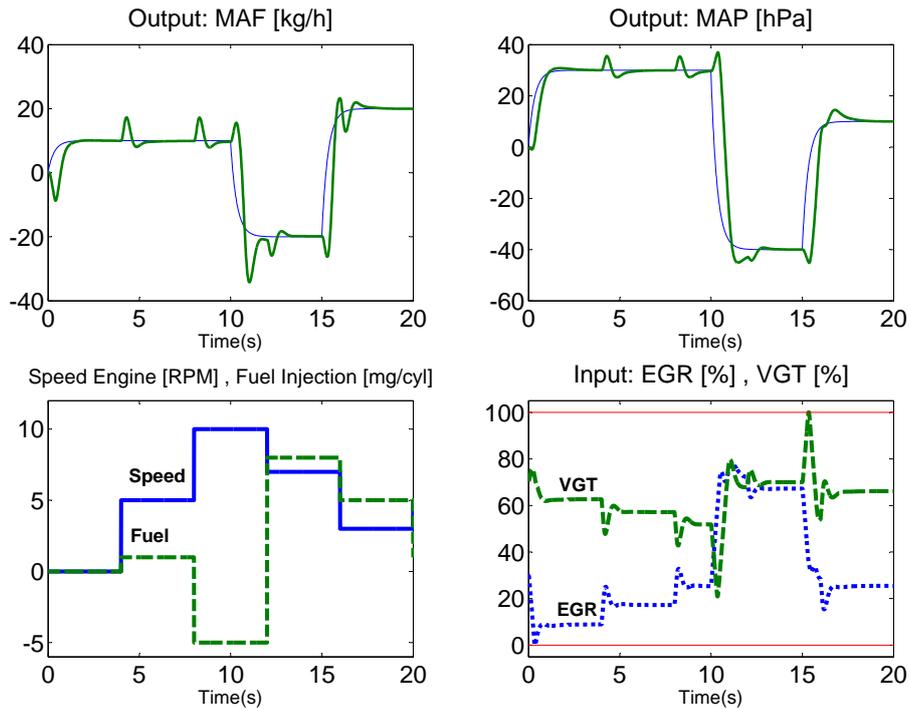


Figure 4.24: Evolution of the closed loop system with the introduction of measured disturbances, engine speed and fuel injection (Model (4.6)).

4.6.2 Simulations Using the Model Depending on the Inputs

When compared to the preceding model, the model used here is of lower order (8 states rather than 14). Moreover, it has been identified using a larger sampling period of 50 ms . Therefore, it can be expected that the fast NMPC scheme proposed here can be easily used in the present configuration since it fitted already the real-time requirement with a higher sampling rate. However, the large sampling period may decrease the quality of the closed-loop performance as the updating rate is slower.

Table 4.4 shows the parameters that were used in simulations:

Table 4.4: Parameters used in simulations - Model (4.7)

Parameter	Value	Expression	Parameter	Value	Expression
N	30	(4.26)	N_o	10	(4.11)
λ	1	(4.25)	q	5	(4.25)
τ_s	0.05s	(4.7)	ρ_x	10^{-4}	(4.26)
ρ_1	1	(4.28)	ρ_2	1	(4.28)
t_r	1s	(4.27)	n_{fe}	30	-
$[u_{min}, u_{max}]^2$	$[0, 100]^2\%$	(4.1)	$[-\delta_{max}, \delta_{max}]^2$	$[-1, 1]^2\%/\tau_s$	(4.2)

The data used for simulations are quite close to those employed in the test bench. The first difference w.r.t the previous section lies in the the measurement disturbances profile w . In the real engine, they are used internally to generate the set points of MAP and MAF from pre-specified look-up tables. Here, a step sequence of fuel injection and engine speed is adopted, as shown in figure 4.25. Remember that the central values w_c are assumed to be 0 according to section (4.2.5.3) which explains the negative values in the figure.

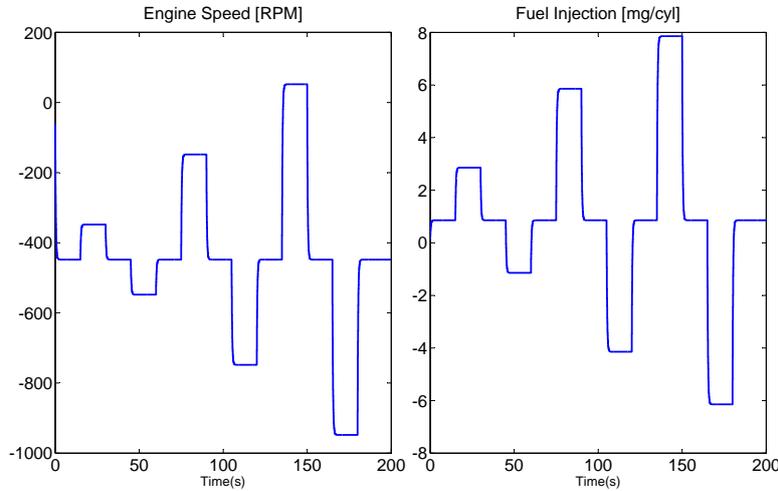


Figure 4.25: Evolution of the measurement disturbances profile (w) used to generate MAF and MAP set-points (y) (Model (4.7)).

Figure 4.26 shows the evolution of the closed loop system under the set-point profiles of MAP and MAF generated by the measurement disturbances w depicted on figure 4.25. It is worth noting that the desired response time of the closed loop t_r (4.27) is set to 1 s, and both, set-points and measurement disturbances are filtered with the same response time. This may enable to avoid or at least reduce overshoots. Moreover, since $\rho_1 = \rho_2 = \bar{y}_1 = \bar{y}_2 = 1$, the outputs $y_1 = \text{MAF}$ and $y_2 = \text{MAP}$ are equally weighted. Consequently, MAP has lower overshoot since its magnitude is

greater than MAF.

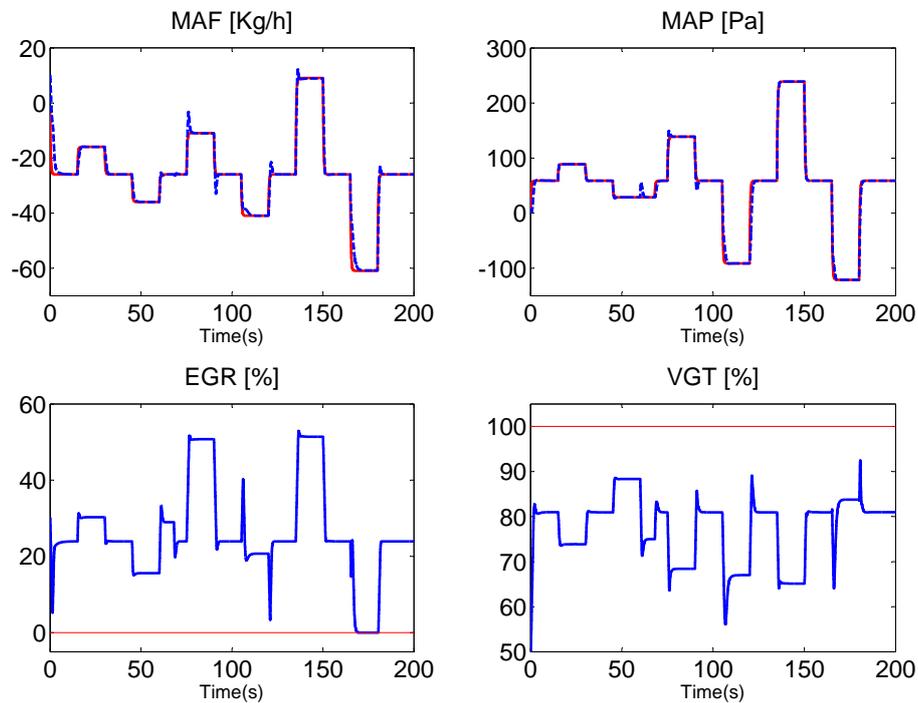


Figure 4.26: Evolution of the closed loop system under the successive steps sequence on MAF and MAP (Model (4.7)).

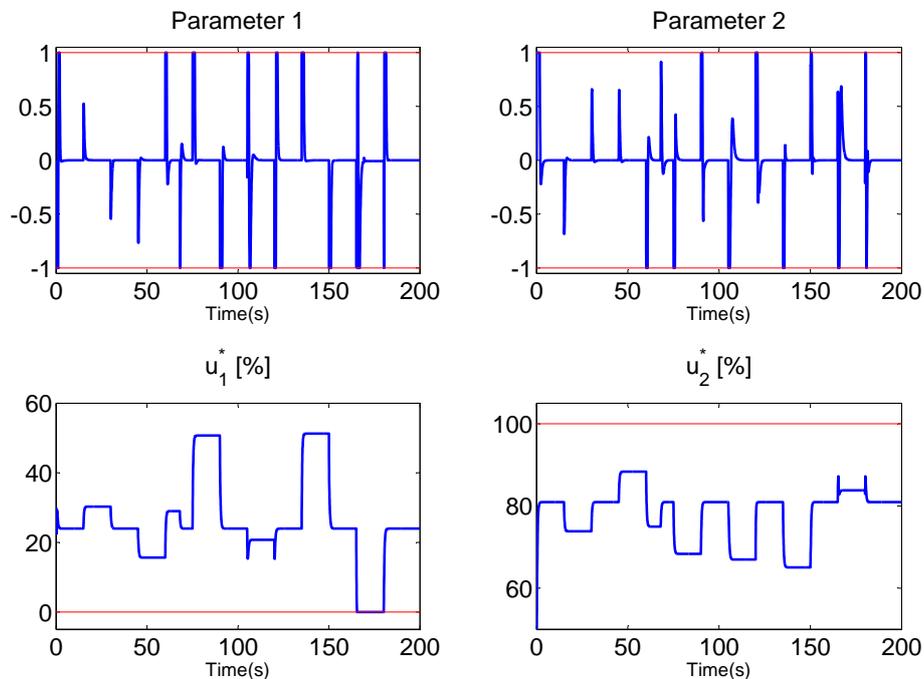


Figure 4.27: Evolution of the set of parameters and the steady control for the scenario depicted on Figure 4.26 (Model (4.7)).

Figure 4.27 shows the evolution of the parameters p_1 and p_2 obtained from (4.26) and the stationary inputs u_1^* and u_2^* calculated from (4.19). Here they are calculated separately in order to obtain x^* but for the general framework both are in the same optimization problem. Note the difference between the steady control profile (u_1^* and u_2^*) and the complete NMPC control strategy with exponential modes shown in the previous scenario (EGR and VGT in Figure 4.26).

4.6.3 Simulations Using the Full Nonlinear Model

As long as simulation results are concerned, the results presented in this section are the most important. This is due to the complexity of the nonlinear model (4.3)-(4.5) that is expected to challenge the real-time implementability of the proposed approach. At the same time, the success of the proposed approach on such a structure-free model would emphasize our claim on the genericity of the proposed approach and its complete independence w.r.t any structural assumptions.

This model is quite recent since it has been delivered to us few months before the beginning of the manuscript writing. This is the reason for which, experimental validation could not be obtained because of the non availability of the test bench and the lack of available time. Nevertheless, the simulation results are quite convincing and incite to push the investigations further.

This model was identified from experimental data on the real diesel engine test bench available at the Johannes Kepler University, Linz. It was delivered to us as MATLAB/SIMULINK block-scheme. The model was totally converted into C code at the University of Grenoble in discrete-time setting resulting in a powerful real-time simulator for diesel engines. Here, the model was sampled at 10 ms using the standard Euler's method.

It is worth noting that the whole state is supposed to be measured by the test bench sensors. This means that the observer is no more needed. Moreover, the cost to be minimized does not take into account x^* leading to the new cost function defined in (4.32). For these reasons, the parameters N_o does not exist in this framework and ρ_x has been set to 0 as explained in section 4.4.5 . The set of parameters are shown in table 4.5.

Table 4.5: Parameters used in simulations - Model (4.3)-(4.5)

Parameter	Value	Expression	Parameter	Value	Expression
N	100	(4.32)	d_p	[0.1;0.5]	(4.33)-(4.34)
λ	8	(4.25)	q	1.25	(4.25)
τ_s	0.01s	(4.3)	ρ_x	0	(4.32)
ρ_1	1	(4.28)	ρ_2	100	(4.28)
\bar{y}_1	1	(4.28)	\bar{y}_2	1e5	(4.28)
t_r	0.1s	(4.27)	n_{fe}	4	-
$[u_{min}, u_{max}]^2$	$[0, 100] \times [50, 100]\%$	(4.1)	$[-\delta_{max}, \delta_{max}]^2$	$[-1, 1]^2\%/\tau_s$	(4.2)

Note that the VGT never works below 50% because in such cases, the compressor losses much power, which is, from a practical point of view, unacceptable. For this reason, the lower bound of 50% is used for the VGT valve position.

The simulation results are divided into two main sets according to the kind of set-point profiles: the sequence of step changes and the New European Driving Cycle (NEDC). The latter is supposed to represent the typical usage of a car in Europe, and is used, among other standardized requirements, to assess the emission levels of car engines.

4.6.3.1 Successive Step Sequence

Figure 4.28 illustrates the closed-loop output behavior under the sequence of step changes on the set-points. Note that this model uses the real values and not their increments w.r.t. some central values. For this reason, the normalization term $\bar{y}_2 = 1e5$ is introduced to fairly penalize both outputs. The tracking performance is quite nice except during the interval time $t \in [165; 180]$ (see MAF) where the corresponding set-points are unfeasible. In this case, the controller does its best to follow the closest output.

It is worth mentioning that the overshoots that can be observed on Figure 4.28 are due to excessively short desired response time which has been set to 0.1s (t_r in (4.27)). Reducing the bandwidth of the filter, the oscillations may reduce considerably. This can be clearly seen in figure 4.29, where the response time was changed to 1s.

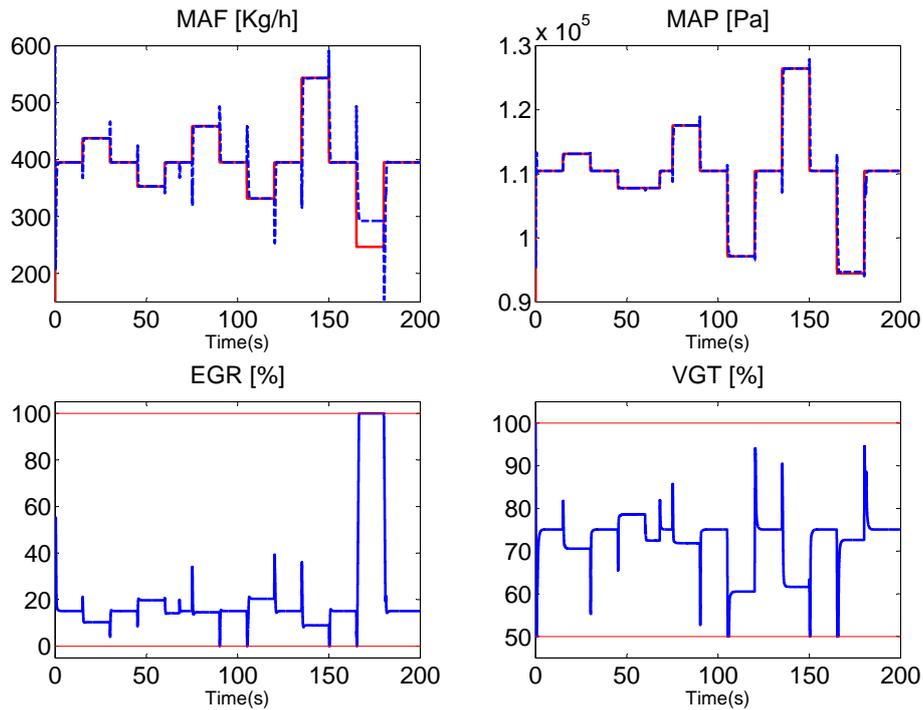


Figure 4.28: Evolution of the closed-loop output under a sequence of step changes and $t_r = 0.1s$. Note that the set-points are unfeasible during $t \in [165; 180]$ and hence the input EGR reaches the saturation in order to find the closest output. (Model (4.3)-(4.5)).

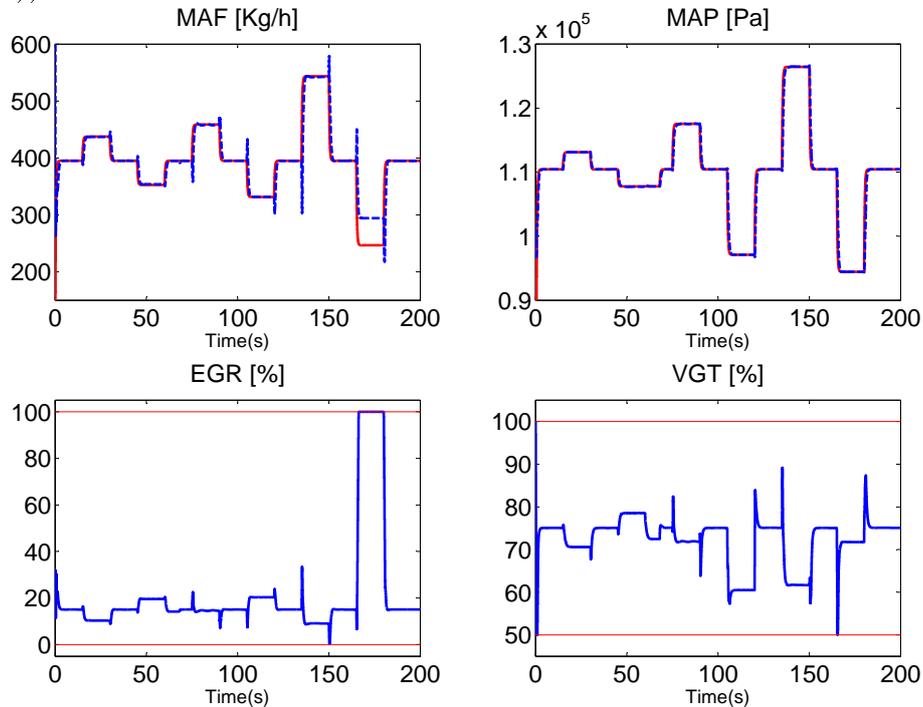


Figure 4.29: Evolution of the closed-loop output under a sequence of step changes and $t_r = 1s$ (Model (4.3)-(4.5)).

Figure 4.30 shows the evolution of the estimation of an offset error on the outputs. The error dynamics are driven by the equation (4.8), which consists in a simple integrator. The integrator gain k_i was set to 3.5 for both outputs. The bias dynamics was modeled as a constant and tries to reproduce the prediction error. This represents the difference between the measured value and the output predicted value. This will be extremely important for real time implementation as shown in the next section.

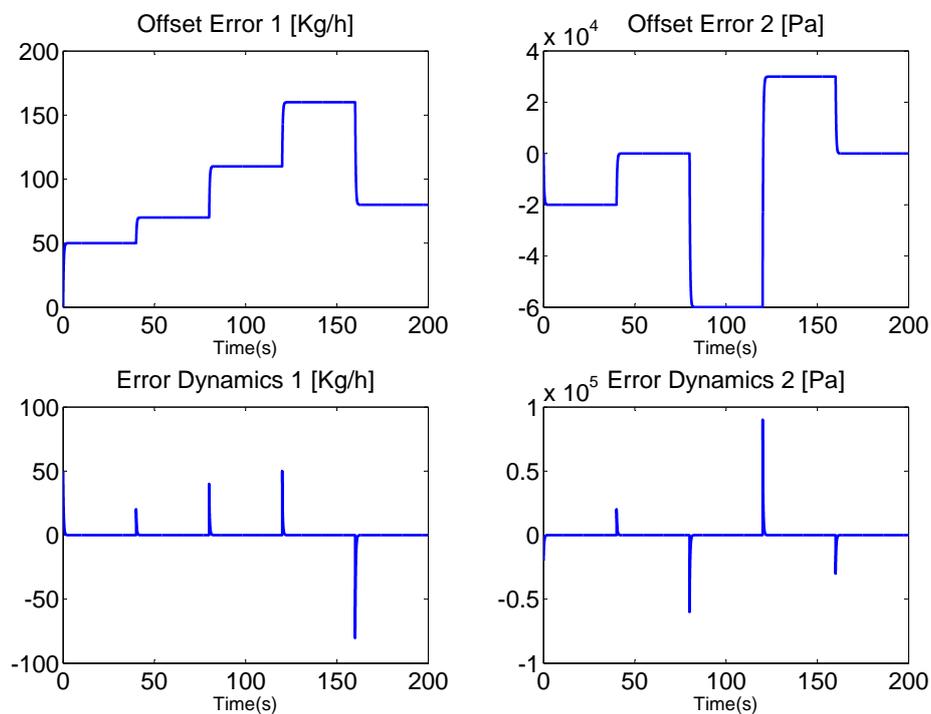


Figure 4.30: Evolution of the offset error estimation and dynamics error between the predicted and measured value (Model (4.3)-(4.5)).

Figure 4.31 shows the evolution of the 4 parameters that are solution of the optimization problem (4.32). Contrarily to the preceding approach where the stationary control was computed separately, here the optimization problem incorporates the steady control value as a supplementary degree of freedom, according to the new set of constraints defined in (4.31).

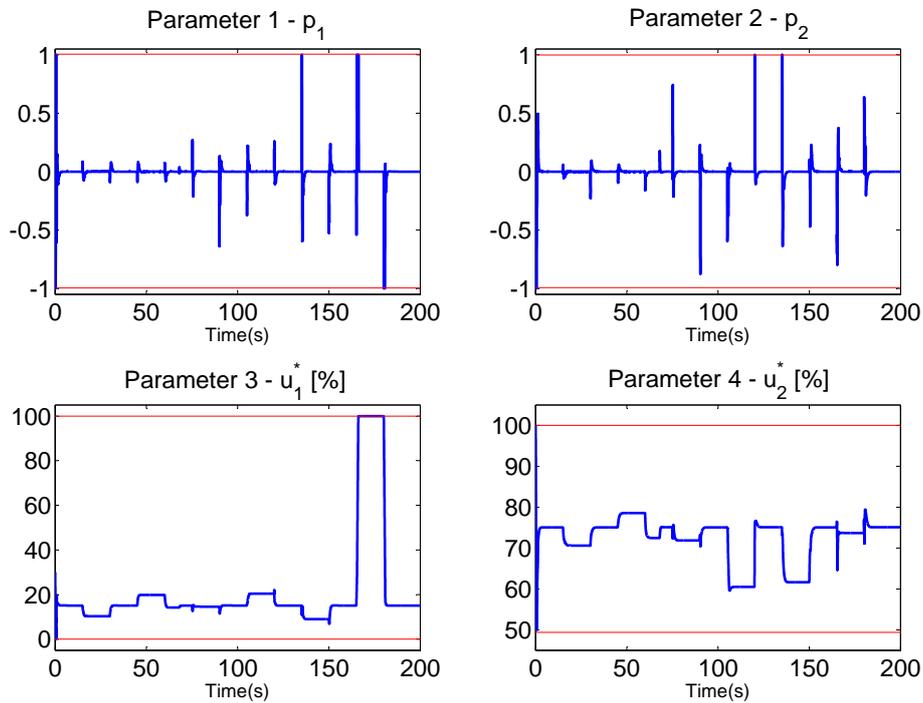


Figure 4.31: Evolution of the 4 parameters used for the optimization problem (4.32), which considers the stationary inputs as a part of the decision variable (Model (4.3)-(4.5)).

Figure 4.32 shows the computation time spent to proceed the fixed number of function evaluations n_{fe} , which in this case, has been set to 4. This represents only 1 SQP iteration according to the algorithm presented in section 4.5.2. Considering that the model is highly nonlinear and constrained, this is a quite interesting result. Indeed, one can perform sufficient number of function evaluations to obtain a fairly good solution in within a sampling period of 10 ms which means that the present NMPC scheme is real-time implementable.

4.6.3.2 New European Driving Cycle - NEDC

In this section, the reference profile to be tracked is the NEDC, which is entirely simulated. This cycle is quite representative in terms of operational conditions. As a matter of fact, it represents a challenging problem for control design since the set-points of MAF and MAP may vary extremely fast and also fall into unfeasible regions.

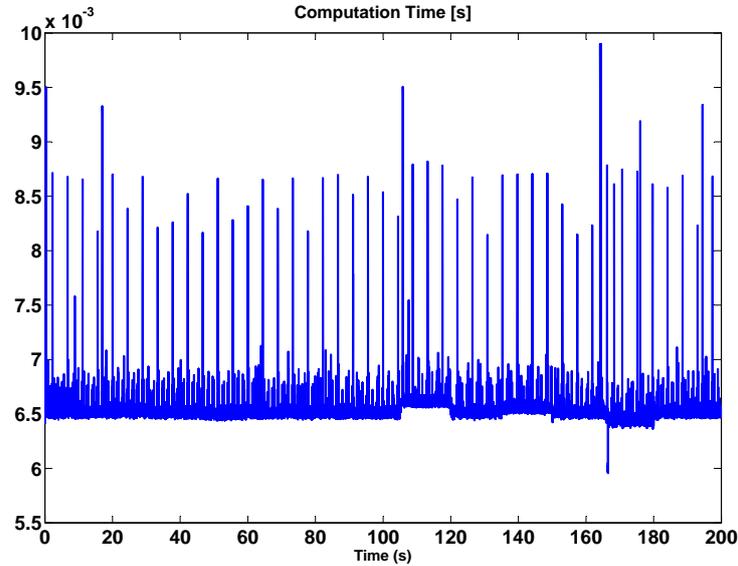


Figure 4.32: Evolution of the computation time of the parameterized NMPC scheme, with $n_{fe} = 4$ which corresponds to only 1 SQP iteration. Note that the maximum allowed value 10 ms is never reached (Model (4.3)-(4.5)).

The NEDC can be divided into two main periods: the urban and extra urban parts, as shown in figure 4.33. The first one is represented by the interval $[0,250]$ s and reproduces fast variations on the speed leading to a high-frequency set-point generation for MAF and MAP. The second which holds on the time interval ranging from 250s to 500s, is slowest but has important excursions.

Figure 4.33 shows the first simulation scenario for the NEDC tracking. The weighting term ρ_2 has been set to 1. Note the good tracking performance in both the urban and extra-urban parts and the constraints on the inputs are correctly handled. Note that the periods where the valves are saturated means that the set of set-points are unfeasible, especially for MAP around 0 s, 100 s and 480 s.

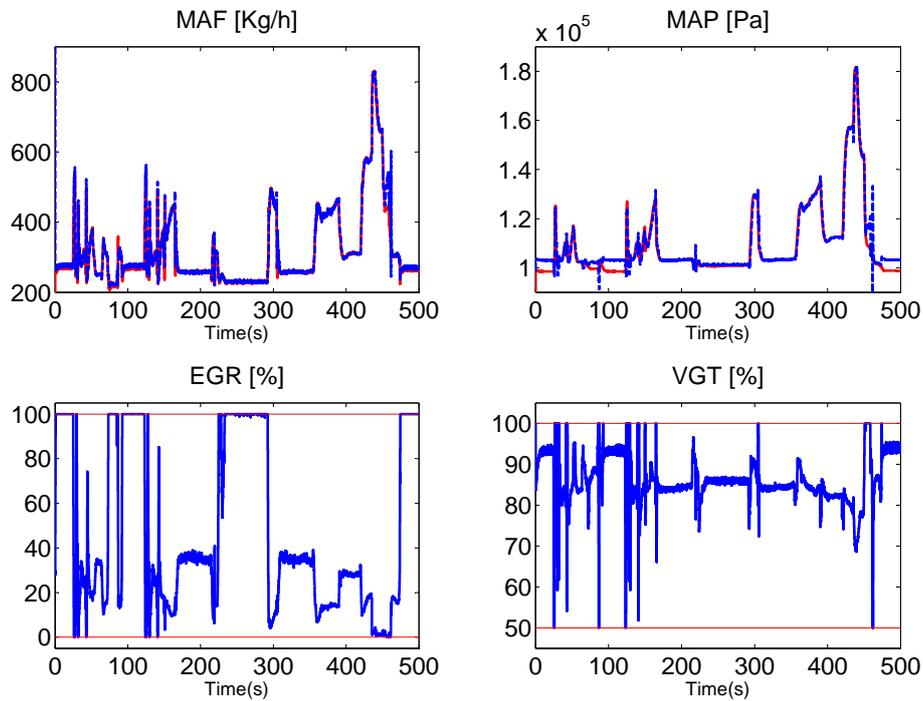


Figure 4.33: Simulation results showing the evolution of MAF and MAP over the whole NEDC tracking, with $\rho_2 = 1$ (4.28). Note the quality of the tracking, especially for the fast set-point variations. Note also the points where the controller is saturated and is not able to handle the output requirements at the same time since the set-points are not feasible (Model (4.3)-(4.5)).

Figure 4.34 illustrates the simulation results for two values of ρ_2 . The first one, $\rho_2 = 100$ and the second one $\rho_2 = 0.01$ (4.28). While in the first case, the tracking performance of MAP is quite nice, for the second one, MAF is privileged. This clearly shows the trade-off between the tracking quality of the set-points and the weighting terms that are imposed on the outputs in the definition of the cost function. Then, such penalizing terms can be modified on-line in order to achieve specific performances in different operational points of the engine according to emissions requirements. Otherwise, one can just find a good trade-off as shown in the previous scenario depicted in figure 4.33.

The next section shows the experimental results obtained when using the model (4.7). As explained previously, the experimental validation using the full non linear model (4.3)-(4.5) was not possible because of the lack of time and availability of the test bench and will be investigated as soon as possible.

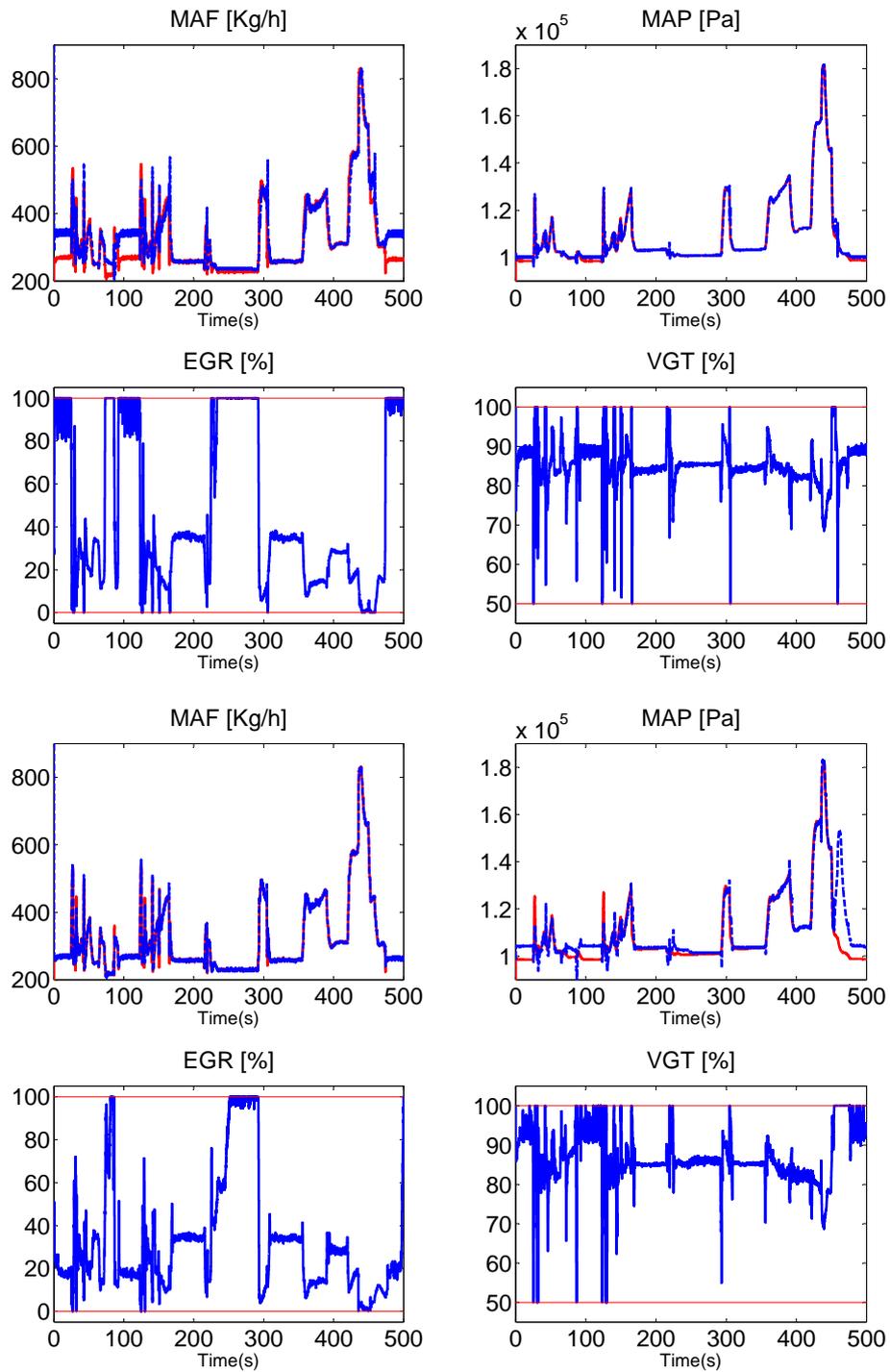


Figure 4.34: Evolution of MAF and MAP over the NEDC tracking, with $\rho_2 = 100$ (on the top) and $\rho_2 = 0.01$ (on the bottom) (Model (4.3)-(4.5)).

4.7 Experimental Results

This section shows the experimental results obtained on the real world diesel engine available at Johannes Kepler University, Linz [Murilo et al., 2009b]. The platform consists in a BMW M47TUE Diesel engine fulfilling the EU4 emission standard, controlled by the Engine Control Unit (ECU) equipped with an AVL dynamo-meter to simulate the load on the engine shaft. A D-SPACE AUTOBOX was used as a real-time software and hardware, running at 480 MHz and linked to the MATLAB software. The programs related to the MHO and the NMPC were developed in C language and transformed into dll files for d-Space. The sampling time was set to 50 ms since the model (4.7) was used for the experimental validation of the controller. Table 4.6 shows the parameters used in the experiments. The technical details about the diesel engine test bench are shown in appendix B.

As in the previous case, two types of experiments are presented hereafter. First, a manual setting of w profile was used, by imposing a step sequence trajectory for both variables. In the second part, the trajectories were generated by the high-speed part of NEDC. Figure 4.35 illustrates the profiles of w used in the experiments. In both cases, the NMPC closed loop performances are compared to the performance of the existing controller driven by the ECU.

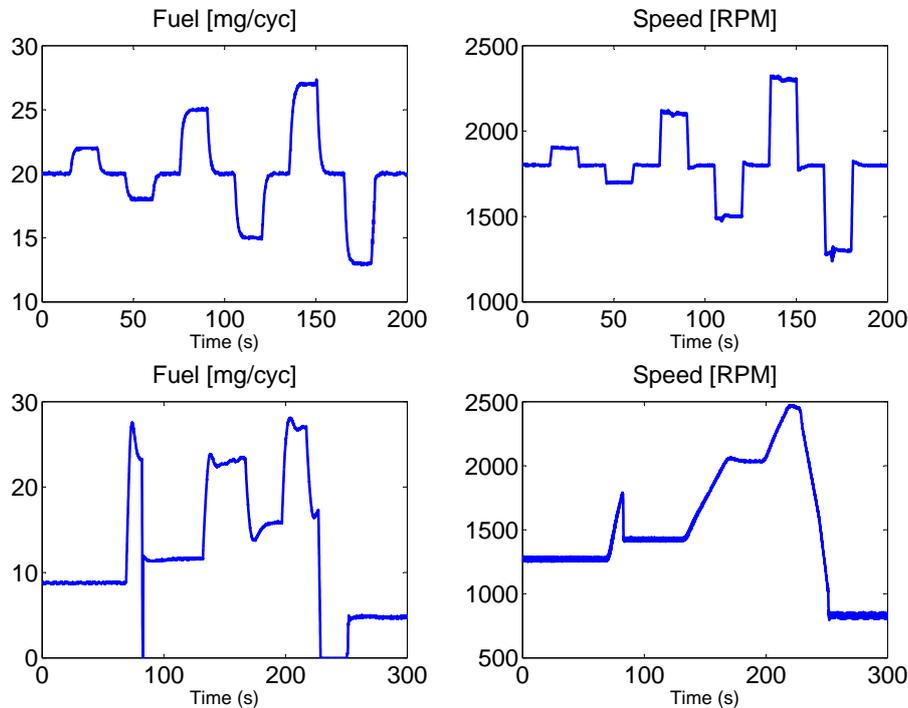


Figure 4.35: The profiles of fuel injection and speed engine used for the set-point generation. On the top the step sequence and on the bottom the non-urban part of the NEDC.

Table 4.6: Parameters used in experiments - Model (4.7)

Parameter	Value	Expression	Parameter	Value	Expression
N	30	(4.26)	N_o	10	(4.11)
λ	1	(4.25)	q	5	(4.25)
τ_s	0.05s	(4.7)	ρ_x	10^{-3}	(4.26)
ρ_1	1	(4.28)	ρ_2	1	(4.28)
t_r	$3 \cdot \tau_s / q \cdot \lambda$	(4.27)	n_{fe}	30	-
$[u_{min}, u_{max}]^2$	$[0, 100]^2\%$	(4.1)	$[-\delta_{max}, \delta_{max}]^2$	$[-1, 1]^2\% / \tau_s$	(4.2)

4.7.1 Successive Steps Sequence

The ECU's response for a step sequence is shown in figure 4.36. The ECU's control design is based on a feed-forward, only for EGR. For VGT, the controller uses a look-up table depending on the operational point of the engine. In this case, the MAF is highly weighted by ECU while the MAP's tracking is considered as a secondary task.

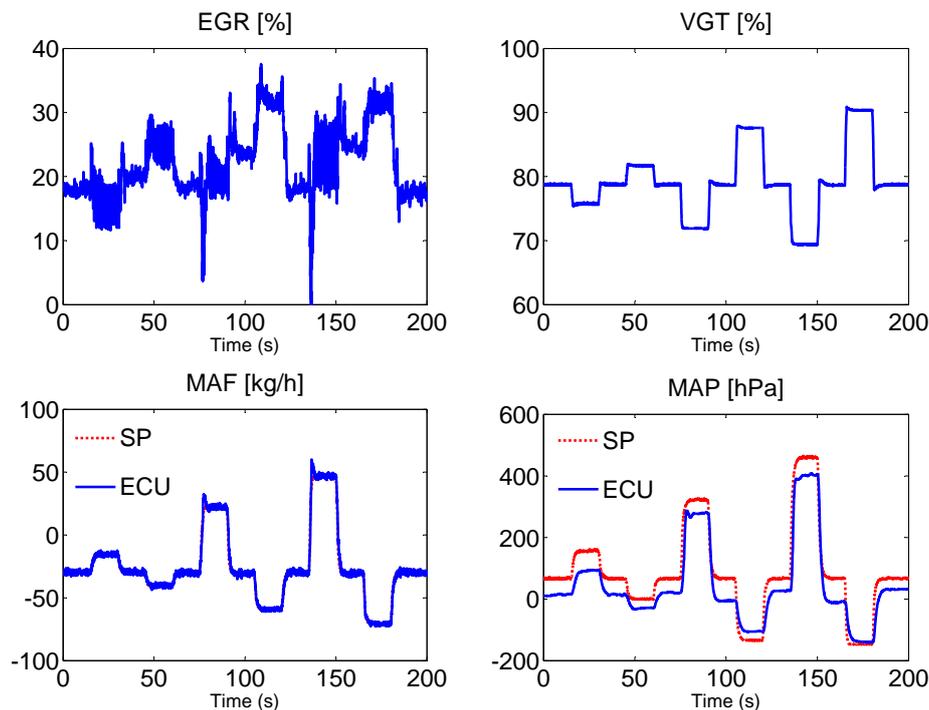


Figure 4.36: ECU's response for a step sequence. Note the difference in the quality of the tracking between the MAF and the MAP.

The first scenario is shown on figure 4.37. The constraints on the inputs are handled, and some overshoots may come from the fact that measured disturbances are not

filtered, specially the engine speed. Both variables are equally weighted and the controller is able to track them correctly. Note the non-minimum phase behavior which forces the inputs to take one direction before finding to the right one. Offset errors are eliminated by means of the integrator term introduced in (4.8) with a integrator gain of $k_i = [0.6; 0.01]$.

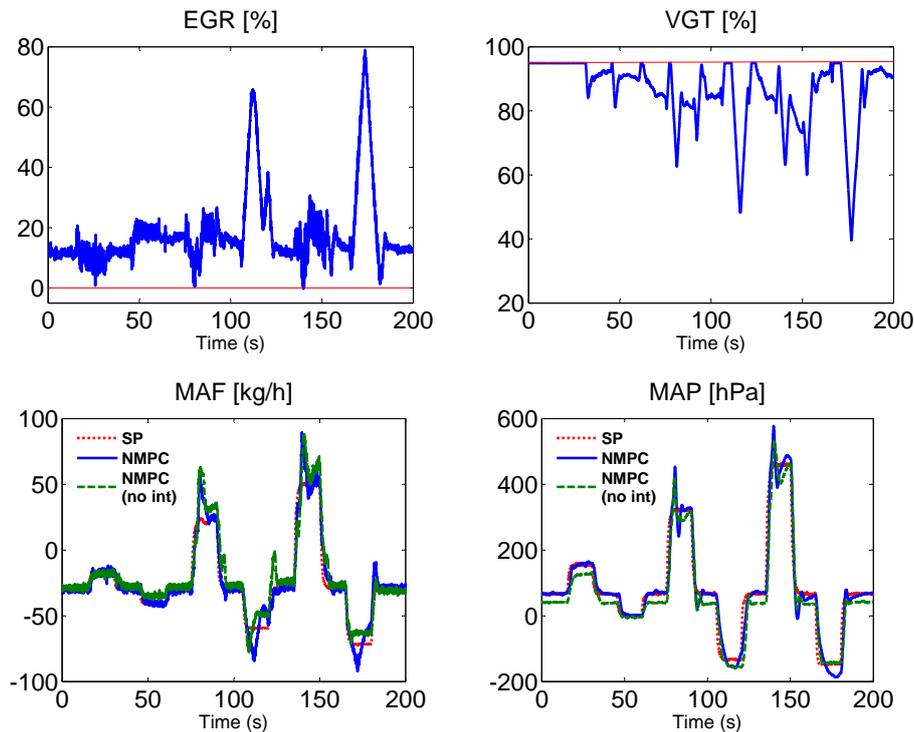


Figure 4.37: Experimental results of the parameterized NMPC scheme on a sequence of steps on the set-points. The green dashed line shows the trajectories without integrator. The MAF and the MAP are simultaneously tracked and the integrator term eliminates the offset error in the stationary state. This figure is to be compared to figure 4.38 where the weight on the terminal state has been increased.

In order to deal with overshoots and oscillations, the terminal state was strongly weighted (parameter ρ_x in (4.26)), as shown in Figure 4.38. However the system's response time becomes a bit slower. This is the classical trade-off between these requirements. Reducing δ_{max} in order to make the controller less aggressive, may also reduce oscillations. Some other scenarios were performed by changing the parameters of table 4.6 without significant improvement on the results.

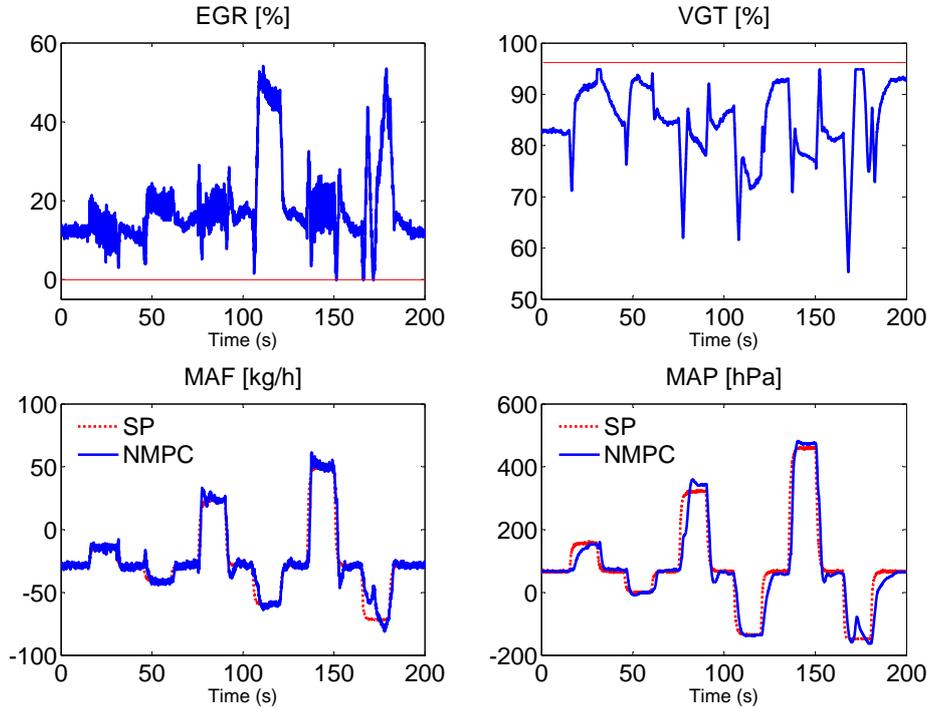


Figure 4.38: Experimental results of the parameterized NMPC scheme, with $\rho_x = 1000$ (4.26). Comparing to the previous scenario, overshoots are reduced by weighting the terminal state at the price of the slower response time.

4.7.2 NEDC Sequence

The second part of the experimental validation consists in testing the present controller under the high-speed part of the NEDC. The top of figure 4.39 shows the ECU's tracking performance. The MAF is correctly tracked until the instant where the engine speed exceeds the 2100 RPM (Fig. 4.35), around $t = 210s$. The same deterioration at high speed is also noted on the MAP's tracking performance.

On the other hand, the results of the parameterized approach presented on the bottom of figure 4.39 clearly shows a more regular quality in the tracking performance for both variables comparing to the ECU's, except in the last part at low speed and fuel. Here, model uncertainties become too high to compensate for. Moreover, the slow dynamics of the engine speed profile of the NEDC does not generate abrupt set-point variations, and naturally filters oscillations and overshoots. However, the present solution seems to be quite sensitive to the fast set-point variations. This is especially due to the fact that the updating rate is not fast enough to compensate for model uncertainties. By using a faster and more reliable model such as the full nonlinear one may expect actual performance level to be significantly improved.

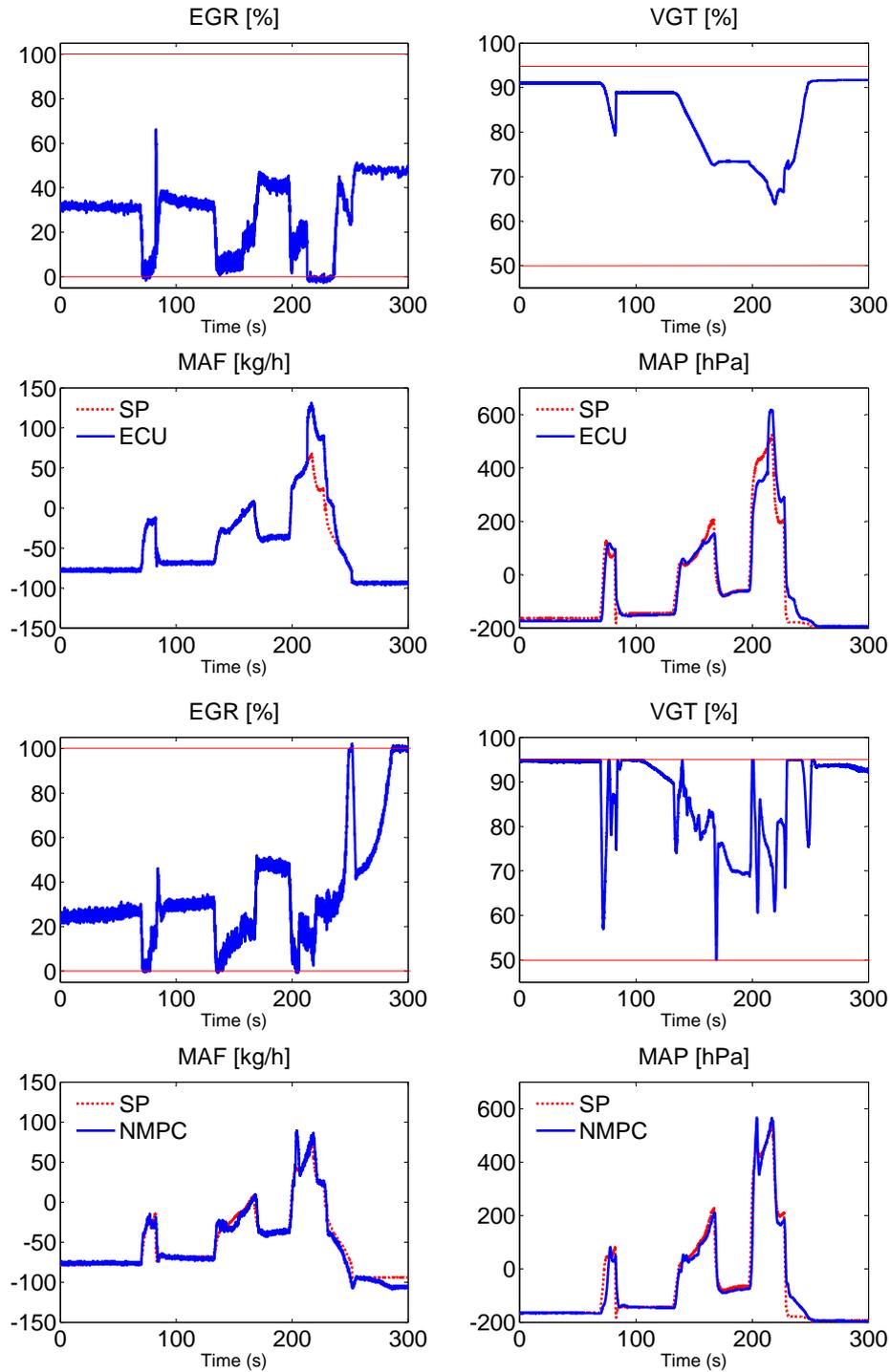


Figure 4.39: Experimental results comparing the ECU and the parameterized NMPC scheme under the extra-urban part of the NEDC. The ECU has a good tracking performance in the beginning but the system becomes very sensitive in the high-speed part. For the NMPC, both variables are correctly tracked, specially in the high-speed part.

4.8 Conclusion

In this chapter, a parameterized NMPC scheme is presented to tackle the diesel engine air path control problem. Simulations were performed with three different nonlinear models, and all of them showed quite nice results. In particular, a highly nonlinear sixth-order model with seven look-up tables was used to test the controller and the results were quite promising. This suggests that the present solution can be used as a sort of black-box NMPC controller for diesel engines, no matter the complexity and the structure of the model being used.

An experimental validation of the present controller was also proposed in this chapter. The experiments performed in a real-world diesel engine of the University of Linz confirmed the efficiency of the controller. Two main scenarios were tested: a sequence of steps and the non-urban part of the NEDC. In both cases, the tracking performance of the MAF and the MAP were very nice and had better results when compared with the ECU. Moreover, the controller structurally satisfied the constraints on the inputs, derivatives and available time to perform optimization problem. However, some tracking problems related to the fast set-point variations were observed. This is mainly due to the model mismatches and the slow sampling period (50 ms). It is more than likely that having better models may considerably increase the tracking performance, especially in the regions of high variations of the set-points.

For this reason, the next step consists in validating experimentally the present controller by means of the full nonlinear model, which is supposed to be much more accurate than the other ones. This validation was not possible due to a lack of availability of the test bench, as explained previously, and we hope to proceed some experiments as soon as the test bench becomes available again.

Chapter 5

NMPC for a Twin-Pendulum System

5.1 Introduction

In this chapter, a parameterized NMPC scheme is presented for the swing-up and the stabilization of a twin-pendulum system under control and state constraints. The basic feature concerns a particular parametrization of the set of candidate control profiles leading to a small vector of decision variable. Experimental results are proposed in order to show the efficiency and the real-time implementability of the proposed NMPC strategy. This results enforce the message of this PhD work: Parameterized NMPC is problem dependent but as long as mechatronic systems are concerned, there is almost always a way to parameterize the decision variable in order to end up with numerically tractable optimization problem.

This chapter is organized as follows. In section 5.2, the inverted pendulum problem is presented. Then, section 5.3 shows the system model and the control problem to be solved is formalized. Section 5.4 concerns the theoretical formulation of the control design, namely the basic idea of the NMPC design scheme is introduced. In section 5.5, a nonlinear observer design is presented in order to estimate potential offset errors that may affect the measurement of the angular positions of the pendulums. In section 5.6, some real-time issues are investigated concerning the control design in order to adapt the theoretical formulation to practical application. Section 5.7 presents some experimental results obtained on a twin-pendulum platform available at the Control Systems Department of the University of Grenoble. Finally, section 5.8 gives the conclusion of this chapter.

5.2 The Inverted Pendulum Problematic

5.2.1 Background on the Existing Variations of the Problem

The inverted pendulum is probably one of the most studied problems by the automatic control community, especially for nonlinear control design. Several control strategies have been developed to tackle the inverted pendulum problematic and many other variations have been created to enlarge the range of possible solutions. Figure 5.1 shows a schematic view of the classical single inverted pendulum problem. The system is represented by a single pendulum fixed on a cart that moves horizontally. The control problem consists in steering the pendulum to the upward at rest position and in keeping it there by acting on the cart's acceleration or force.

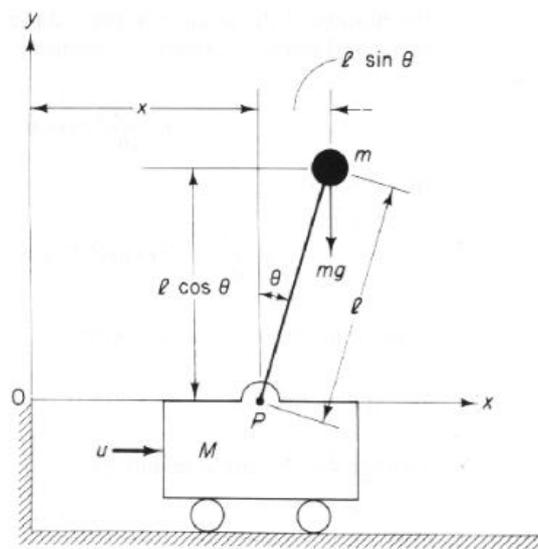


Figure 5.1: Classical single inverted pendulum problem [Ogata, 2002].

One of the classical variations derived from the above original problem is the use of two pendulums, instead of a single one. One possible scenario is shown in figure 5.2, also called the double pendulum system. It is a coupled system whose control problem is still the same, to steer to and to keep both pendulums at the upward vertical position.

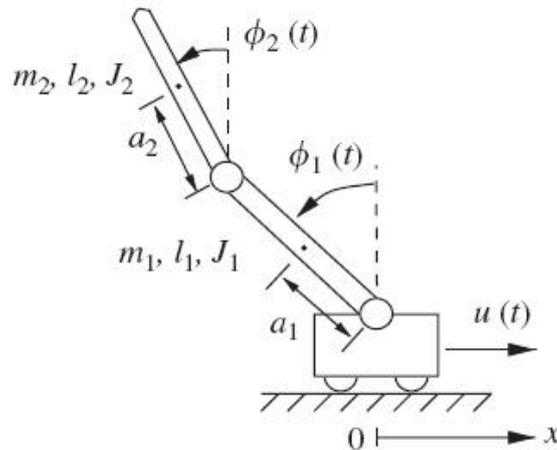


Figure 5.2: Schematic view of a double-pendulum system [Graichen et al., 2007].

Another widely used variation is the rotary inverted pendulum, also known as the Furuta pendulum shown in figure 5.3. The main difference in this case is the fact that the base is rotative, and the control input is the angular acceleration or couple. Moreover, there exist configurations with a double-pendulum system as explained above, but in this case, with a rotative base.

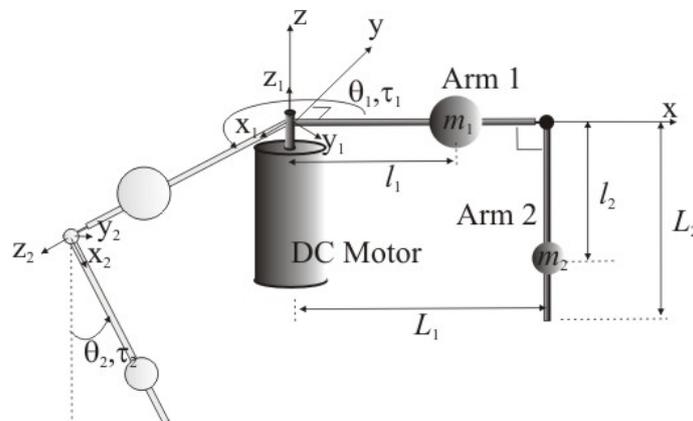


Figure 5.3: Representation of the Furuta rotary inverted pendulum.

Note that if the pendulums are in the neighborhood of the equilibrium point, linear control design tools may be sufficient. However, considering that the pendulum starts from a different initial state, the resulting problem becomes much more difficult to solve. In this context, the term *swing-up* systems is invoked, since a swing-up phase is needed in order to bring the pendulum to the neighborhood of the upward vertical at rest position. The main difficulty presents in such configurations is the constraint on the cart's position, for the horizontal case depicted on Figure 5.1. For

this reason, the swing-up phase for the rotary inverted pendulum is much easier to realize than in the horizontal case, since there is no constraints on the angular position of the motor to be respected.

Some relevant works can be found in the literature concerning the global stabilization of swing-up like systems [Åström and Furuta, 2000, Chatterjee et al., Zhong and Röck, 2001, Graichen et al., 2007]. Nevertheless, the configuration presented in this PhD work falls in a more specific variation of the inverted pendulum problem, namely, the twin-pendulum system, which is described in the next section.

5.2.2 The Twin-Pendulum System

The twin-pendulum system is a variant of the configuration used for two pendulums presented in figure 5.2. Actually, the main difference with the double-pendulum system lies in the fact that the pendulums are completely decoupled from each other, and both are connected to the cart. Figure 5.4 shows the twin-pendulum platform that was developed in the Control Systems Department, Grenoble, for the experimental validation of the proposed parameterized NMPC strategy described in this chapter.

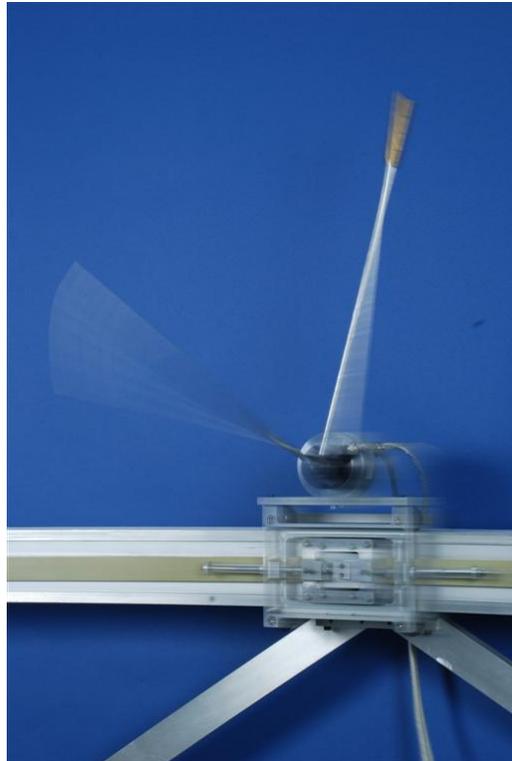


Figure 5.4: Twin-pendulum system of Control Systems Department (University of Grenoble) used as experimental platform.

In this case, the two pendulums must have different inertias to make the system controllable and the constraints on the cart's position must be respected, since the base of the platform is linear and horizontal. Therefore, the resulting control problem becomes not only quite nonlinear but also under-actuated and highly constrained.

As a matter of fact, the presence of state constraints that are related to the limited excursion of the cart makes this problem extremely challenging. This may explain the relatively limited number of papers found in the literature to deal with such a system. In [Xin and Kaneda, 2005, Åstrom and Furuta, 2000], an energy based method is presented in order to make the energy level of the pendulums converge to the desired value. Moreover, a Lyapunov approach is used together with a stability analysis based on the concept of invariant sets. However, the constraints on the cart's position are not considered.

Another control strategy was presented in [Bortoff, 1996], but in this case, with a rotary twin-pendulum configuration. Figure 5.5 shows a schematic view of this system. In this approach, a state feedback linearization approach is used and the stability of the resulting zero dynamics is analyzed. Basically, the strategy consists in two parts. The first part is dedicated to the stabilization of the smallest of the two pendulums and the second part to stabilize the biggest pendulum by using local oscillations of the first around the upward desired position. The properties around the vertical upward position was invoked by [Lundberg and Roberge, 2003]. Nevertheless, since it is a rotational system, the constraints on the angular position of the motor are not present and the control problem is hence much easier to tackle.

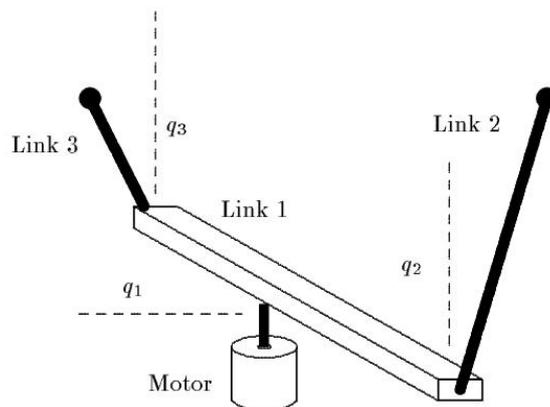


Figure 5.5: Rotary twin-pendulum presented in [Bortoff, 1996] without constraints on the angular position of the steering devices.

From the preceding discussion, it comes that NMPC-like solutions may be particularly adapted to the twin-pendulum problem, not only to deal with the system nonlinearities but also to handle the constraints. Moreover, since the previous ap-

proaches tackle the problem in a continuous time setting, this necessarily leads to a small sampling period, and hence the computation time becomes also an issue. For this reason, a dedicated parameterized NMPC strategy is proposed here in order to swing-up and to stabilize the twin-pendulum system at the upward vertical at rest position under the constraints mentioned above. But first, the mathematical model of the system needs to be presented. This is the aim of the next section.

5.3 System Modeling

The schematic view of the twin-pendulum platform used in our experiments is shown in figure 5.6. This system consists in two pendulums with different inertias, completely decoupled and fixed on a cart. In this figure the notations m_i, l_i, I_i are used to refer to masses, lengths and moments of inertia respectively while θ_i stands for the angular position of the i_{th} pendulum, r the cart's position measured from the origin O_o and r_{max} the maximum admissible value for r . The force F is applied on the cart generated by the control input u , which is the torque delivered by the motor.

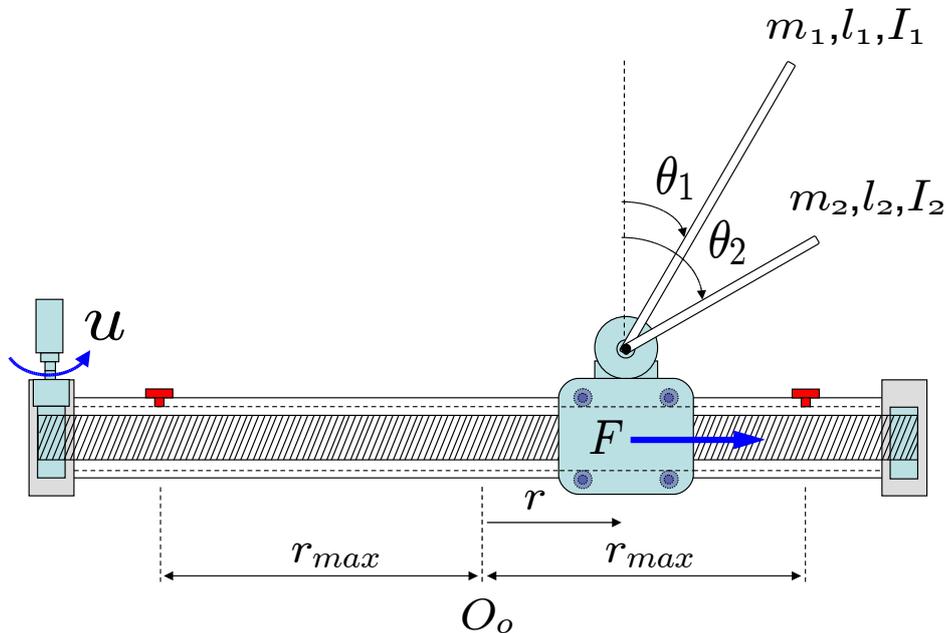


Figure 5.6: Twin-pendulum system (Control Systems Department, Grenoble).

The mathematical model of the system consists basically in two main parts: the

mechanical model and the DC motor model. Both are naturally linked and they generate a unique model for the whole system. The description of each one of these dynamics are detailed in the next section.

5.3.1 The Model of the Cart/Pendulum System

The mechanical model of the cart and the pendulum can be derived using the Lagrange equations:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = Q \quad (5.1)$$

where q is the vector of generalized coordinates given by :

$$q = [r \quad \theta_1 \quad \theta_2]^T$$

and Q represents the vector of the generalized forces acting on the system. Recall that the Lagrangian of the system is given by:

$$L = T - P \quad (5.2)$$

where T and P represent respectively the kinetic and potential energies. The kinetic energy of the system is the sum of the kinetic energy of each of its components, namely the cart, the first and the second pendulum. Denoting these energies by T_0 , T_1 and T_2 respectively, one clearly has:

$$\begin{aligned} T_0 &= \frac{1}{2} m_0 \dot{r}^2 \\ T_1 &= \frac{1}{2} m_1 \left[(\dot{r} + l_1 \dot{\theta}_1 \cos \theta_1)^2 + (l_1 \dot{\theta}_1 \sin \theta_1)^2 \right] + \frac{1}{2} I_1 \dot{\theta}_1^2 \\ T_2 &= \frac{1}{2} m_2 \left[(\dot{r} + l_2 \dot{\theta}_2 \cos \theta_2)^2 + (l_2 \dot{\theta}_2 \sin \theta_2)^2 \right] + \frac{1}{2} I_2 \dot{\theta}_2^2 \end{aligned} \quad (5.3)$$

and the potential energies of each component are given by:

$$P_0 = 0 \quad ; \quad P_1 = m_1 g l_1 \cos \theta_1 \quad ; \quad P_2 = m_2 g l_2 \cos \theta_2 \quad (5.4)$$

Then, from (5.3) and (5.4), the Lagrangian L can be obtained according to (5.2) as follows:

$$\begin{aligned} L &= \frac{1}{2} (m_0 + m_1 + m_2) \dot{r}^2 + \frac{1}{2} (m_1 l_1^2 + I_1) \dot{\theta}_1^2 + \frac{1}{2} (m_2 l_2^2 + I_2) \dot{\theta}_2^2 + \\ &\quad m_1 l_1 \dot{r} \dot{\theta}_1 \cos \theta_1 + m_2 l_2 \dot{r} \dot{\theta}_2 \cos \theta_2 - m_1 g l_1 \cos \theta_1 - m_2 g l_2 \cos \theta_2 \end{aligned} \quad (5.5)$$

In the absence of friction, by developing the equation (5.1) and (5.5), the resulting system becomes:

$$\begin{aligned} F &= (m_0 + m_1 + m_2)\ddot{r} + m_1 l_1 \ddot{\theta}_1 \cos \theta_1 - m_1 l_1 \dot{\theta}_1^2 \sin \theta_1 + \\ &\quad + m_2 l_2 \ddot{\theta}_2 \cos \theta_2 - m_2 l_2 \dot{\theta}_2^2 \sin \theta_2 \\ 0 &= (m_1 l_1^2 + I_1)\ddot{\theta}_1 + m_1 l_1 (\cos \theta_1)\ddot{r} - m_1 g l_1 \sin \theta_1 \\ 0 &= (m_2 l_2^2 + I_2)\ddot{\theta}_2 + m_2 l_2 (\cos \theta_2)\ddot{r} - m_2 g l_2 \sin \theta_2 \end{aligned}$$

Or equivalently:

$$\ddot{\theta}_1 = -\alpha_1 \ddot{r} \cos \theta_1 + \beta_1 \sin \theta_1 \quad (5.6)$$

$$\ddot{\theta}_2 = -\alpha_2 \ddot{r} \cos \theta_2 + \beta_2 \sin \theta_2 \quad (5.7)$$

$$F = \alpha_0(\theta_1, \theta_2)\ddot{r} + \beta_0(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \quad (5.8)$$

where the following notations are used

$$\alpha_i = \frac{m_i l_i}{m_i l_i^2 + I_i} \quad ; \quad \beta_i = g \alpha_i \quad ; \quad i \in \{1, 2\} \quad (5.9)$$

$$\alpha_0(\theta_1, \theta_2) = (m_0 + m_1 + m_2) - \alpha_1 m_1 l_1 \cos^2 \theta_1 - \alpha_2 m_2 l_2 \cos^2 \theta_2 \quad (5.10)$$

$$\beta_0(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = m_1 l_1 \sin \theta_1 (\beta_1 \cos \theta_1 - \dot{\theta}_1^2) + m_2 l_2 \sin \theta_2 (\beta_2 \cos \theta_2 - \dot{\theta}_2^2) \quad (5.11)$$

Note that in order to complete these equations, one needs to link the force F needed to move the cart to the torque developed at the motor level. This is introduced in the following section.

5.3.2 The Model of the DC Motor

For practical implementation, it is necessary to introduce the classical dynamics of a DC motor which is used to deliver the force F needed to move the cart. In order to produce the desired force F , the motor has to deliver a torque that is related to the force F by the following differential equation:

$$(N_r^2 J_m + J_a)\ddot{r} + (N_r^2 k_{vm} + k_{va})\dot{r} + N_r r_p C_s = N_r r_p u - r_p^2 F \quad (5.12)$$

where N_r is the reduction ratio, J_m , J_a the moments of inertia of the motor and axis respectively, k_{vm} and k_{va} the viscous friction coefficients at the motor and the axis respectively, r_p the radius of the pulley and C_s the resistant dry torque on the shaft axis. These values are provided by the manufacturer and are shown in appendix C.

Based on the above, the following expression can be obtained by injecting the expression of F (5.8) in (5.12):

$$(N_r^2 J_m + J_a + r_p^2 \alpha_0) \ddot{r} + (N_r^2 k_{v_m} + k_{v_a}) \dot{r} + N_r r_p C_s + r_p^2 \beta_0 = N_r r_p u$$

leading to the classical modeling for mechanical systems of the form:

$$M_e \ddot{r} + C_e \dot{r} + K_e = u$$

where

$$\begin{aligned} M_e &= \frac{N_r^2 J_m + J_a + r_p^2 \alpha_0(\theta_1, \theta_2)}{N_r r_p} \\ C_e &= \frac{N_r^2 k_{v_m} + k_{v_a}}{N_r r_p} \\ K_e &= \frac{N_r r_p C_s + r_p^2 \beta_0(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)}{N_r r_p} \end{aligned}$$

5.3.3 The Complete Model

Putting together the above sub-models, the resulting control input u becomes:

$$u = \alpha_0^m(\theta_1, \theta_2) \ddot{r} + \beta_0^m(\dot{r}, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$$

where $\alpha_0^m = M_e$ and $\beta_0^m = C_e \dot{r} + K_e$ are the modified values of the inertias α_0 (5.10) and β_0 (5.11). To summarize, the system equations, derived from (5.6) and (5.7), becomes:

$$\begin{cases} \ddot{\theta}_1 = -\alpha_1 v \cos \theta_1 + \beta_1 \sin \theta_1 \\ \ddot{\theta}_2 = -\alpha_2 v \cos \theta_2 + \beta_2 \sin \theta_2 \\ \ddot{r} = v \end{cases} \quad (5.13)$$

where v is the cart's acceleration. Therefore, the corresponding control input u can be defined as follows:

$$u = \alpha_0^m(\theta_1, \theta_2) v + \beta_0^m(\dot{r}, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \quad (5.14)$$

Once the model is defined, the next section states the control problem.

5.3.4 The Control Problem

The aim of the present work is to design a control law that stabilizes the following set:

$$\{x \in \mathbb{R}^6 | x = (r^f \ 0 \ 0 \ 0 \ 0 \ 0)^T; r^f \in [-r_{max}, +r_{max}]\} \quad (5.15)$$

where x is the state. Note in particular that the final position of the cart is not imposed but has to be in the admissible range $[-r_{max}, r_{max}]$. Moreover, this control law must respect the constraints on the input and position during the closed-loop evolution. This can be written as follows:

$$r(t) \in [-r_{max}, +r_{max}] \quad (5.16)$$

$$u(t) \in [u_{min}, u_{max}] \quad (5.17)$$

The control design presented here consists in a hybrid controller which switches between two distinct phases: the swing-up phase, the most important one for the purpose of this PhD work since it is driven by the parameterized NMPC scheme, and the linear phase, where a standard local controller stabilizes the pendulums at the upward at rest vertical position. The control design is presented in the following section.

5.4 Parameterized NMPC

5.4.1 Model Analysis

First of all, let us examine some fundamental properties of the model. These properties concern the concept of energy orbit that has been introduced by [Åstrom and Furuta, 2000]. This will be quite important in the following development in order to define the contractive scheme used for the NMPC design.

The energy orbit of a pendulum i can be defined by:

$$E_i = \frac{1}{2}\dot{\theta}_i^2 + \beta_i(\cos\theta_i - 1) \quad ; \quad i \in \{1, 2\} \quad (5.18)$$

the time derivative of E_i can be easily obtained, taking into account the system (5.13):

$$\dot{E}_i = -\alpha_i v \dot{\theta}_i \cos\theta_i \quad ; \quad i \in \{1, 2\} \quad (5.19)$$

Based on this last expression, the following four important properties can be derived assuming for the time being that the friction forces are totally absent:

- During time intervals where $v = 0$, the quantity E_i is constant. The two pendulums evolve on a constant energy orbit. Moreover, if at some instant, one has $E_i = 0$, then the future trajectory under $v = 0$ leads to the state $\theta_i = \dot{\theta}_i = 0$ which represents the rest desired vertical position.
- As soon as one has $\dot{\theta}_i \cos \theta_i \neq 0$, there is non zero gain between the control and E and one can act on the derivative of E in order to steer incrementally E to 0 as shown below.
- On the contrary, during time intervals where $\dot{\theta}_i \cos \theta_i \approx 0$, E_i is insensitive to v .
- The intuition behind the proposed solution is that the time periods during which $(\dot{\theta}_1 \cdot \cos \theta_1)$ and $(\dot{\theta}_2 \cdot \cos \theta_2)$ are both equal to zero are quite rare (because the pendulums properties are by assumption different) and therefore, one can use periods where one pendulum is weakly sensitive to v to *improve* the energy orbit level of the other.

Based on the above intuition, a contractive NMPC scheme can be proposed to solve the control problem during the swing-up phase.

5.4.2 The Control Design

First of all, let us define the following normalized energy orbits according to:

$$E_i^n = \frac{|E_i|}{E_i^{max}} \quad ; \quad i \in \{1, 2\} \quad (5.20)$$

where $E_i^{max} = 2\beta_i$ is the value of E_i when $\theta_i = 180^\circ$ and $\dot{\theta}_i = 0$, namely, at the downward vertical at rest position. These two normalized orbit values are then used to define the overall system's energy orbit according to:

$$E = \max\{E_1^n, E_2^n\} \quad (5.21)$$

The aim of the NMPC scheme used during the swing-up phase is to steer the energy orbit of the overall system, namely E , to 0. Note that by definition, when $E = 0$, $E_1 = 0$ and $E_2 = 0$ which means that both pendulums evolve on orbits that end at the vertical upward at rest position.

In order to formalize the NMPC design, it is necessary to address the constraints, namely the constraints on the inputs and on the state (position of the cart). In order to address the constraint on the cart's position, a change of control variable is adopted. More precisely, the desired position r_d is used as a new decision variable for the NMPC design. The original control input is computed such that the cart position tracks the desired position r_d (the new decision variable). This gives the following dynamic change in the control variable:

$$v(k) = -K_s \begin{pmatrix} r(k) - r_d(k) \\ \dot{r}(k) \end{pmatrix} \quad (5.22)$$

where $K_s \in \mathbb{R}^{1 \times 2}$ is a stabilizing feedback gain matrix. By doing so, one can meet the constraint on the cart position provided that the following conditions are respected:

1. The set-point r_d is chosen such that $r_d \in [-r_{max}, +r_{max}]$
2. The feedback gain is computed such that the transient response starting from zero velocity and from some initial admissible position never leaves the admissible range $[r_{min}, r_{max}]$.
3. The changes in the set-point occurs at instant where \dot{r} is almost equal to 0.

Note also that under the above conditions, one can always tune the stabilizing gain K_s such that the resulting control trajectories never exceed the saturation levels. In fact, there are two ways to satisfy the constraints on the input:

- Monitoring the maximal allowable increment on r_d , defined hereafter as δ_r^{max} .
- Designing the gain K_s for a given δ_r^{max} , which leads to closed-loop trajectories on u that do not violate the constraint. To do this, a simple pole placement of the discrete representation of (5.22) is used, and the desired discrete poles $h_i \in \mathbb{R}^2$ of the closed-loop can be chosen as follows:

$$h_i = e^{-5\tau_s/t_r} \quad , \quad i \in \{1, 2\} \quad (5.23)$$

where τ_s is the sampling period and t_r the desired settling time of the closed loop of the cart position associated to the pre-compensator (5.22). Then, for a given τ_s , the gain K_s can be tuned by acting on t_r . In other words, the tuning of this pre-compensation loop enables the saturation constraint (5.17) to be respected provided that the NMPC design uses only sequences of r_d involving increments that are lower than δ_r^{max} .

Roughly speaking, when t_r is increased [resp. when δ_r^{max} is decreased], the maximum value of the control v , and hence u (5.14), decrease. The simulation results presented in the forthcoming sections, will clearly show the relationship between t_r (and hence K_s), δ_r^{max} and the saturation levels of the control input.

Let us use some sampling period $\tau_s > 0$. Denote by $x(k)$ the state of the system at the sampling instant k . Let us adopt a piecewise constant profile for r_d namely

$$r_d(k\tau_s + t) = r_d(k) \quad ; \quad t \in [0, T_p] \quad (5.24)$$

where T_p is the prediction horizon $T_p = N\tau_s$.

Hereafter, it is assumed that the prediction horizon T_p is taken greater than the response time t_r . More precisely, this condition ensures that at the end of the prediction horizon, one has $v \approx 0$ which guarantees (in the absence of friction) that when the prediction horizon is shifted, the final value of the orbit level E can be kept constant by keeping the same value of the desired position. This is crucial for the stability of the NMPC scheme. Figure 5.7 illustrates the behavior of E under the condition $T_p \geq t_r$, when some step change d_r is applied to the set-point r_d .

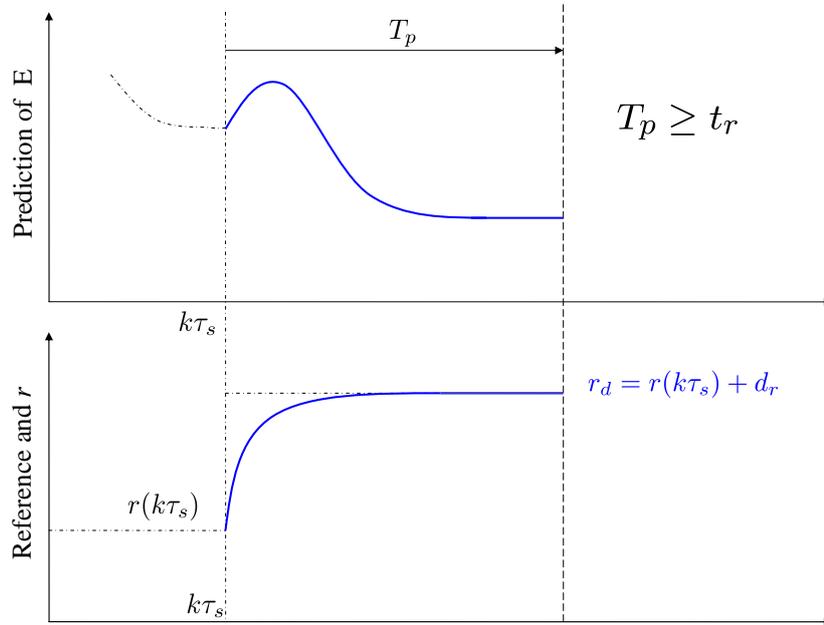


Figure 5.7: Schematic view of the predicted evolution of E over the prediction horizon when $T_p \geq t_r$.

Let us define the following cost function that is related to the state of the system at the end of the prediction horizon:

$$J(x(k), r_d) = E(F(x(k), T_p, r_d))$$

where $F(x(k), T_p, r_d)$ is the solution of (5.13) at instant T_p , starting from $x(k)$ and under the control law defined by (5.22). Referring to the parameterized NMPC scheme developed in the preceding section, the parameter to be optimized here in order to compute the control profile over the prediction horizon is precisely given by the value of $p = r_d$. This leads to the following cost function in the decision variable p :

$$J(x(k), p) = E(F(x(k), T_p, p)) \quad (5.25)$$

Note that the decision variable p has to meet the constraint on the cart position, namely $p \in [-r_{max}, +r_{max}]$.

The next step consists in finding a suitable optimization strategy to provide the optimal parameter of p . Standard optimization routines may be too heavy to be executed in a fraction of the sampling time. Here, the properties of the function E described before are used to formulate the optimization problem in a particularly efficient way. The idea is to check three possibilities:

- the one consisting in keeping the present position by taking $r_d = r$,
- the one consisting in taking a step to the left by using:

$$r_d = \max\{-r_{max}, r - d_r\}$$

where d_r is some step size to be discussed later.

- the one consisting in taking a step to the right by using

$$r_d = \min\{+r_{max}, r + d_r\}$$

Note that the step size d_r need to be smaller when the system reaches the desired orbit since a precise tracking is needed while it can be higher when the system is far from the orbit. That is the reason why the step size d_r is taken as a function of the orbit level E (see figure 5.8) according to:

$$d_r(E) = d_{r_0} + \delta_r E \quad (5.26)$$

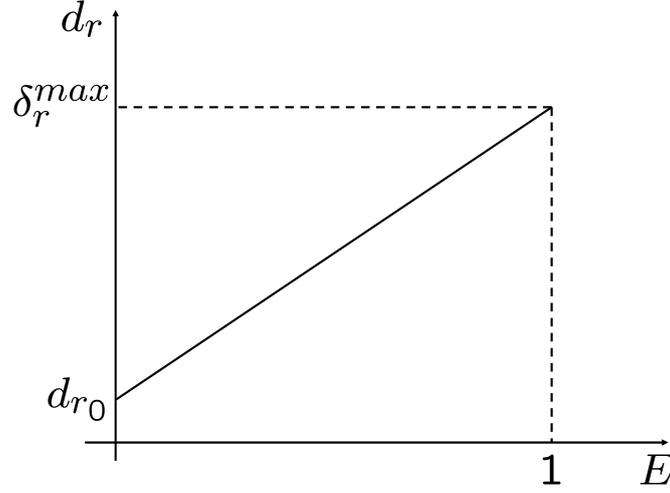


Figure 5.8: Definition of the allowable increment d_r of the cart position as a function of the energy level E . Note that the maximum value δ_r^{max} occurs at $E = 1$ in order to respect the constraints on the inputs. The step d_r decreases according to E up to the minimum value d_{r_0} .

which means that for $E = 1$ the maximum allowable value for d_r is applied, which means δ_r^{max} , and as long as E decreases, the value of d_r decreases until the minimum value d_{r_0} is reached as shown in figure 5.8.

Therefore, the admissible set of the decision variable p would be given by the following discrete set:

$$\mathbb{P} := \left\{ \max\{-r_{max}, r - d_r\}, r, \min\{+r_{max}, r + d_r\} \right\} \quad (5.27)$$

Note however that by using this admissible set \mathbb{P} , it is possible that the swing-up phase ends with r being equal to an extremal value, namely $-r_{max}$ or r_{max} . This would be quite problematic since the final phase (based on a linear controller design in the neighborhood of the desired state) still need some maneuvers that can be made impossible starting from that extremal position (see Figure 5.9). That is the reason why we define a *virtual stops* by introducing the quantity:

$$r_{max}^{RHC} < r_{max}$$

that is used as a virtual stops for the Receding-Horizon Control design. Consequently, the admissible set of the decision variable p becomes:

$$\mathbb{P} := \left\{ \max\{-r_{max}^{RHC}, r - d_r\}, r, \min\{+r_{max}^{RHC}, r + d_r\} \right\} \quad (5.28)$$

Now, the NMPC design can be defined. Namely, at each decision instant, the optimal parameter \hat{p} is obtained by solving the following optimization problem:

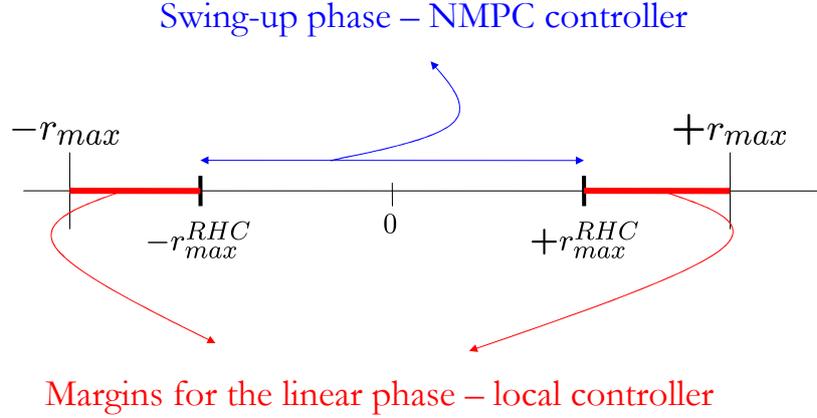


Figure 5.9: Definition of the admissible range for the swing-up phase and the margins for the linear phase. The virtual stops $-r_{max}^{RHC}$ and r_{max}^{RHC} are used by the NMPC controller during the swing-up phase in order to be sure that when the controller switches to the linear control, there is still room for the corresponding maneuvers without hitting the real stops.

$$\hat{p}(x(k)) = \arg \min_{p \in \mathbb{P}} J(x(k), p) \quad (5.29)$$

and since the admissible set \mathbb{P} is a discrete set which contains only 3 elements, the complexity is reduced to the integration of the system equations 3 times, one for each value of p in \mathbb{P} . Figure 5.10 shows an intuitive view illustrating how these 3 choices of p affects the behavior of E and the corresponding optimum values of p that decrease the orbit level.

Injecting the optimal value $\hat{p}(x(k))$ in the expression of the control law (5.22), one obtains the expression of the predictive control law:

$$v(k) = K_{RHC}(x(k)) = -K_s \begin{pmatrix} r(k) - \hat{p}(x(k)) \\ \dot{r}(k) \end{pmatrix} \quad (5.30)$$

In order to complete the definition of the control strategy, the following needs to be defined:

1. A control law to be used in the neighborhood of the desired state,
2. A switching condition that enables to clearly define when to switch between the two controllers

This is explained in the following section.

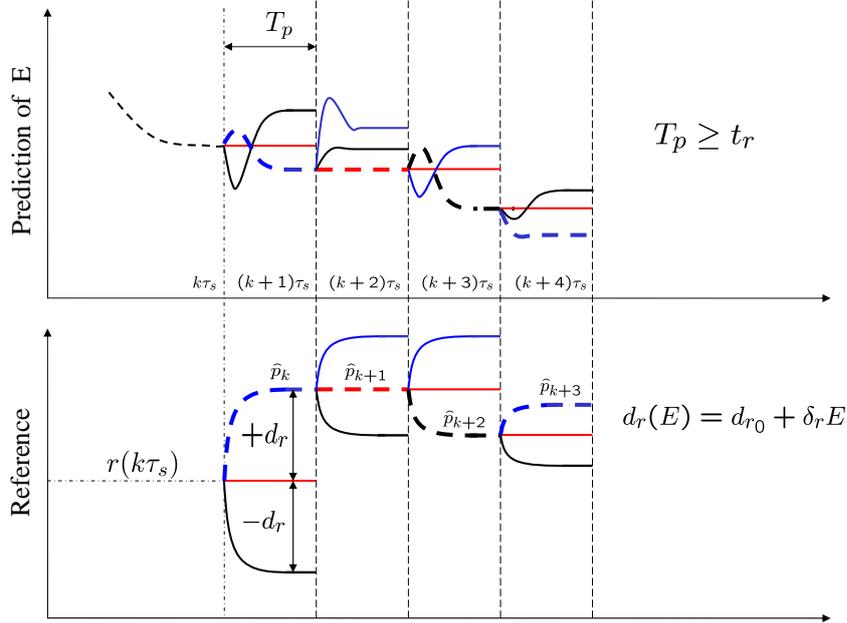


Figure 5.10: Intuitive view showing the contraction of E according to the optimum values of \hat{p} . Top Figure: The dashed lines indicate the trajectories leading to the minimum value of E at the end of the prediction horizon. Bottom Figure: The optimal open loop trajectories of the cart position driven by the parameter \hat{p} . Note that as long as E decreases, the maximum allowable increment d_r decreases too in accordance with (5.26).

5.4.3 The Hybrid Controller

Here, the linearized model of system (5.13) around $x = 0$ is considered. The new system equations become:

$$\begin{cases} \ddot{\theta}_1 = -\alpha_1 v + \beta_1 \theta_1 \\ \ddot{\theta}_2 = -\alpha_2 v + \beta_2 \theta_2 \\ \ddot{r} = v \end{cases} \quad (5.31)$$

which clearly describes a linear system such that:

$$\dot{z} = Az + Bv \quad (5.32)$$

where z is the new state variable $z = (\dot{r} \ \theta_1^m \ \dot{\theta}_1 \ \theta_2^m \ \dot{\theta}_2) \in \mathbb{R}^5$, and $\theta_1^m, \theta_2^m \in [-\pi, +\pi]$ are equal to θ_1, θ_2 , modulo 2π . Note that since the final position r^f is not

imposed, only the velocity \dot{r} is involved in the linear feedback controller design. The matrices A and B are clearly given by:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \beta_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \beta_2 & 0 \end{pmatrix} ; \quad B = \begin{pmatrix} 1 \\ 0 \\ \alpha_1 \\ 0 \\ -\alpha_2 \end{pmatrix}$$

For the above system, a standard linear control strategy can be used. For instance, a discrete-time linear quadratic regulator can be designed to derive the feedback gain matrix K_L to be used in the expression:

$$v(k) = -K_L \cdot z(k) \quad (5.33)$$

The above control law locally asymptotically stabilizes the upward vertical at rest position. Therefore, it can be used only in a small neighborhood of the final desired state. In order to define the region where this controller can be fired, it is usual to use the following characterization that is based on the level set of the corresponding Lyapunov function:

$$\mathcal{A}_0 = \{z \in \mathbb{R}^5 | z^T S z \leq \rho_0\} \quad (5.34)$$

where ρ_0 is some small value usually defined as $\rho_0 = z_0^T S z_0$. In this case, z_0 represents the state that defines the boundaries around the vertical position where linear assumptions are more valid. Therefore, the linear controller (5.33) cannot be fired as long as the state z remains outside \mathcal{A}_0 .

However this condition is not sufficient to switch the controller from the NMPC to the linear law because of the presence of state constraint on the cart position. This is because the transient trajectory of the cart during the application of the linear controller may not satisfy the constraint on the position, as shown in figure (5.11).

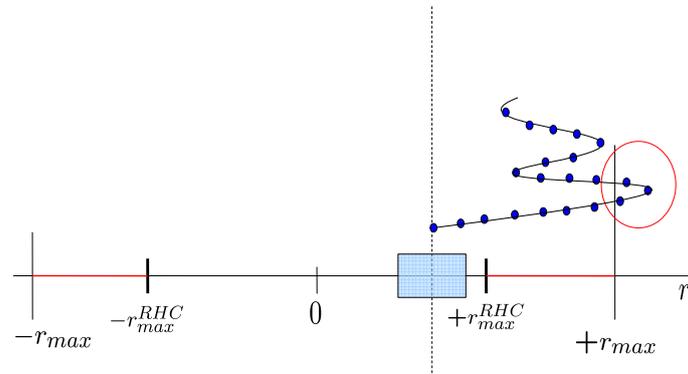


Figure 5.11: The transient of the closed-loop trajectory under the linear controller (5.33) may not respect the constraint on the cart position.

For this reason, a second condition must be added to express the fact that all the positions that would be visited during the transient phase under the linear controller must be within the admissible range $[-r_{max}, +r_{max}]$. This can be expressed by:

$$\varphi(z(k)) \leq r_{max} \quad (5.35)$$

where

$$\varphi(z(k)) := \max_{i \in \{1, \dots, N\}} \left| r(k) + \tau_s(1 \dots 1) \begin{pmatrix} (C(A_d - B_d K_l)^1 z(k)) \\ \vdots \\ (C(A_d - B_d K_l)^i z(k)) \end{pmatrix} \right| \quad (5.36)$$

where $C = (1 \ 0 \ \dots \ 0) \in \mathbb{R}^5$ is the matrix that selects the first element of a 5-dimension vector, which means the cart velocity. Indeed, the term $(A_d - B_d K_l)^1 z(k)$ represents the state on the closed-loop trajectory of the linearized model at instant $(k + i)\tau_s$ and hence $C(A_d - B_d K_l)^1 z(k)$ is the velocity. If such terms are added and multiplied by the sampling time τ_s and then added to the current position $r(k)$, the result may give a precise approximation of the positions that would be visited by the cart during the transient under the linear controller.

To summarize, the conditions that need to be checked before the linear controller is fired are given by:

$$\text{(state close to the target)} \quad z(k) \in \mathcal{A}_0 \quad (5.37)$$

$$\text{(safe transient)} \quad \varphi(z(k)) \leq r_{max} \quad (5.38)$$

The hybrid controller can finally be defined as follows:

$$v(k) = \begin{cases} -K_L z(k) & \text{if } z^T(k) S z(k) \leq \rho_0 \text{ and } \varphi(x(k)) < r_{max} \\ K_{RHC}(x(k)) & \text{otherwise} \end{cases}$$

5.4.4 Simulation Results

This section presents some simulation results involving the control design presented so far. The simulations are divided into two parts. The values used for simulations of the first part are shown in table 5.1. The system starts at the downward position, which means initial state $x_0 = (0 \ 0 \ \pi \ 0 \ \pi \ 0)^T$. To better visualize the evolution of θ_1 and θ_2 , the unit was set to π radians, then, the upward vertical position are represented by the even values in the graphic. The parameter ρ_0 is computed

according to $\rho_0 = z_0^T S z_0$ where $z_0 = (0 \ \pi/6 \ 0 \ \pi/6 \ 0)^T$. To simplify analyzes, the parameters of the DC motor were not considered and the cart acceleration v was used as control input.

Table 5.1: Parameters used in the first part of simulations

Parameter	Value	Parameter	Value
T_p	1.0 s	t_r	0.4 s
r_{max}^{RHC}	0.4 m	r_{max}	0.8 m
τ_s	0.1 s	d_{r_0}	0.1 m
α_1	0.67 m^{-1}	α_2	1.33 m^{-1}

The first simulations are depicted on the figures 5.12 and 5.13. Both illustrate the two angular positions θ_1 and θ_2 , the cart's position, the energy level, and the acceleration v . Note that the NMPC strategy successfully steers the orbit level to 0, leading both pendulums around the vertical position, and the linear controller definitely stabilizes the whole system with the cart being in the admissible range $[-r_{max}, +r_{max}]$.

The only difference between these two scenarios is the parameter δ_r^{max} . For the first one, $\delta_r^{max} = 0.4$ while for the second one, $\delta_r^{max} = 0.1$. Note the difference between the acceleration levels represented by $v = \ddot{r}$. In fact, when δ_r^{max} increases, faster achievement of the task may be realized at the price of higher control v . Therefore, the constraints on the inputs can be handled by tuning the parameter δ_r^{max} . Moreover, the change in the control variable proposed in (5.22) guarantees that the constraints on the position are always respected.

In this second part of simulations, the values used for experimental validation were tested. The set of parameters used in this part are shown in table 5.2.

Table 5.2: Parameters used in the second part of simulations

Parameter	Value	Parameter	Value
T_p	0.6 s	δ_r^{max}	0.4 m
r_{max}^{RHC}	0.6 m	r_{max}	0.7 m
τ_s	0.05 s	d_{r_0}	0.1 m
α_1	0.67 m^{-1}	α_2	1.33 m^{-1}

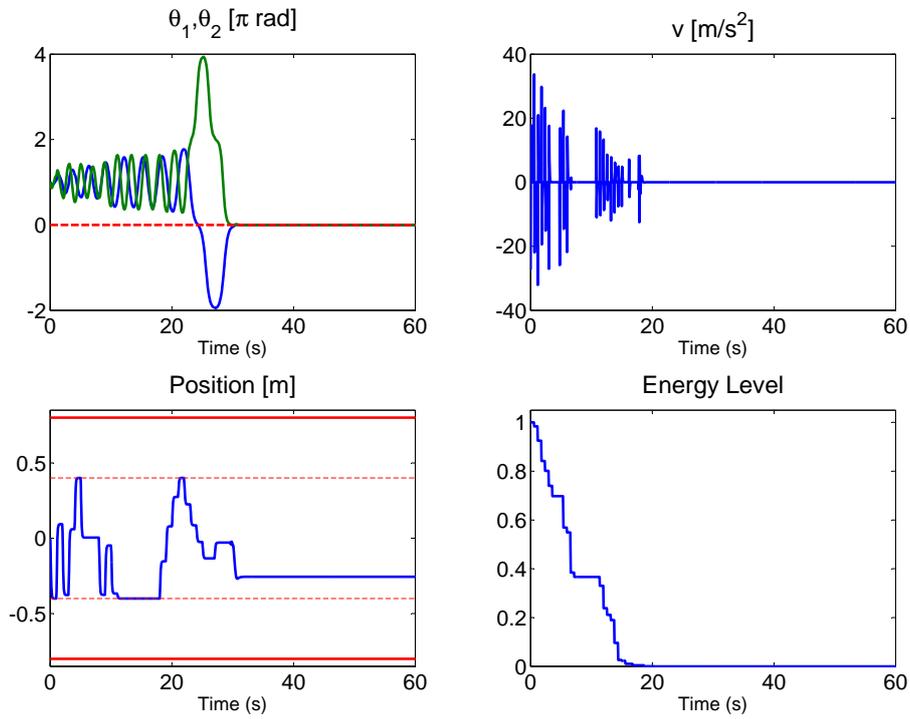


Figure 5.12: Evolution of the closed-loop system under $\delta_r^{max} = 0.4m$ (Table 5.1).

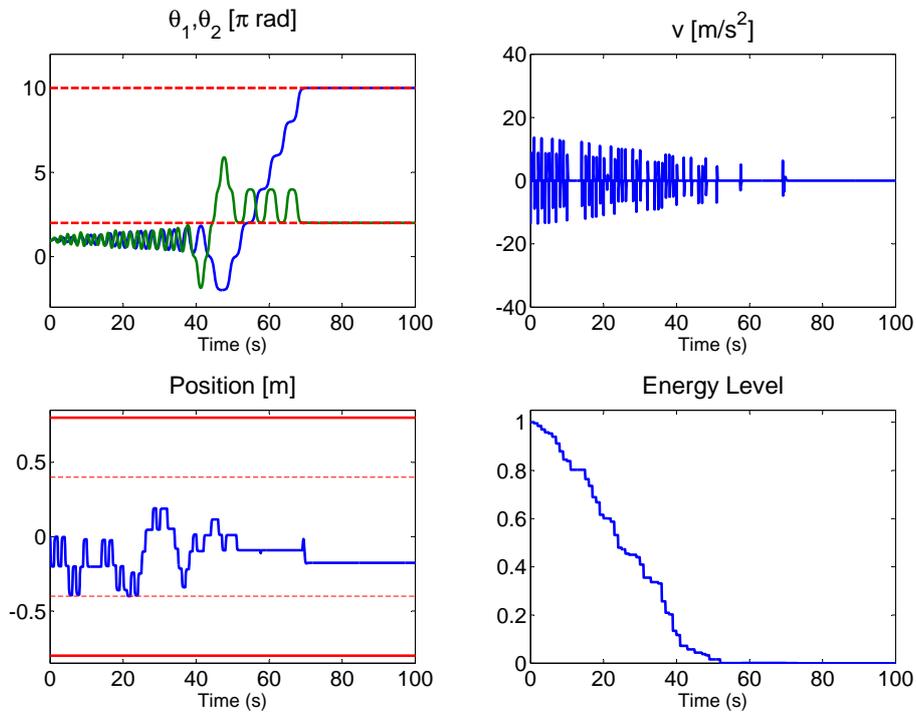


Figure 5.13: Evolution of the closed-loop system under $\delta_r^{max} = 0.1m$ (Table 5.1).

The first scenario of this part is shown in figure 5.14. Note that when the cart position is saturated at $-r_{max}^{RHC}$, no improvements can be made in the orbit level, which remains constant since friction is not considered. Here, the desired response time is $t_r = 0.1s$, which considerably increases the acceleration level v .

In the second scenario, the response time t_r was set to $t_r = 0.6s$. The acceleration level is drastically reduced when compared to the previous scenario (figure 5.14). Moreover, the response time was taken equal to the prediction horizon T_p . As a matter of fact, this condition $t_r = T_p$ is used for real-time implementation and it is explained in section 5.6. Note that even in this limit case, the controller is still able to decrease the orbit level and stabilize the pendulums at upward vertical at rest position.

The last scenario shows the evolution of the closed-loop system starting from a different initial condition $x_0 = (0 \ 0 \ 120 \text{ deg} \ 1.5 \text{ rad/s} \ -100 \text{ deg} \ -2 \text{ rad/s})$. Note that the starting value of E is lower than 1 since its maximum value is the downward vertical at rest position (5.20).

Note that all the developments explained above require the knowledge of the state vector. Assuming that only the angular positions of the pendulums and the position of the cart are accessible to measurement. This means that an observer design is needed for this nonlinear system. The design of such an observer is explained in the next section.

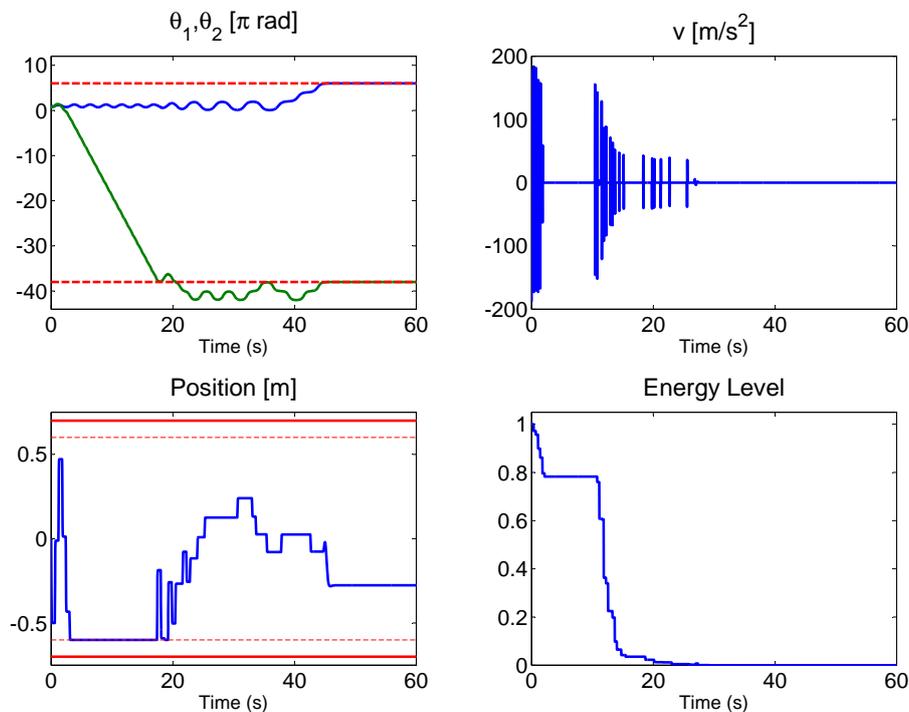


Figure 5.14: Evolution of the closed-loop system under $t_r = 0.1s$ (Table 5.2).

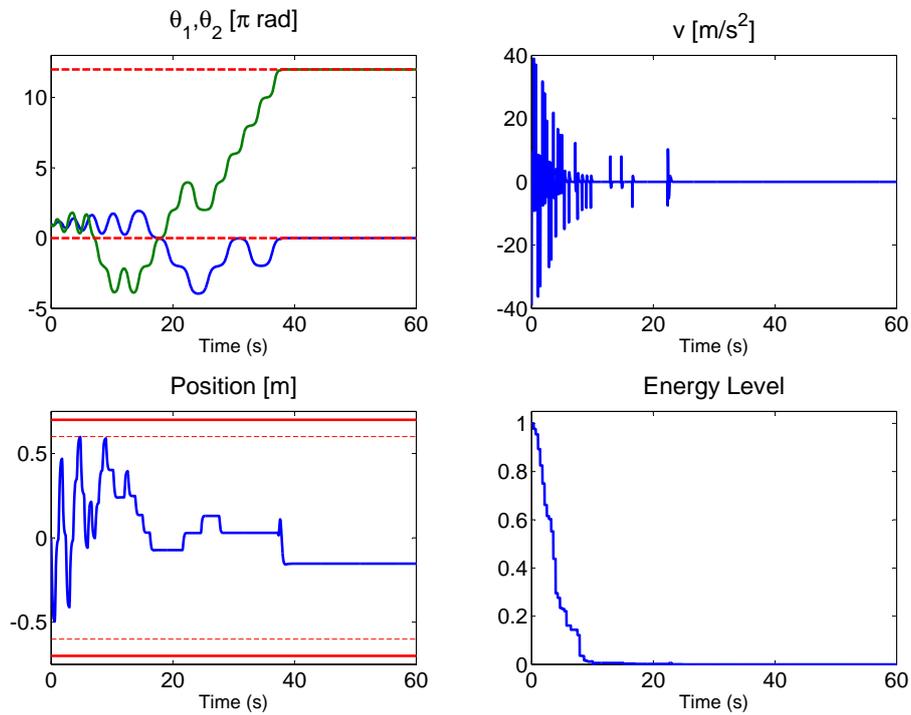


Figure 5.15: Evolution of the closed-loop system under $t_r = T_p = 0.6s$ (Table 5.2).

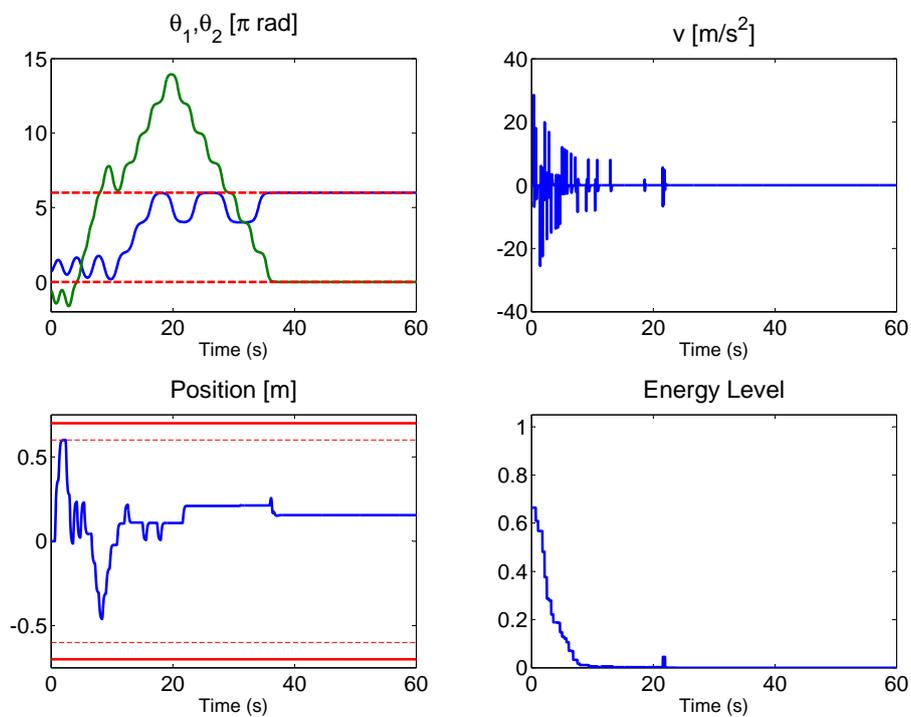


Figure 5.16: Evolution of the closed-loop system starting from the initial state $x_0 = [0, 0, 120 \text{ deg}, 1.5 \text{ rad/s}, -100 \text{ deg}, -2 \text{ rad/s}]$ (Table 5.2).

5.5 Observer Design

5.5.1 Introduction

In the previous section, a parameterized NMPC formulation has been proposed in order to stabilize the vertical upward at rest position of the twin-pendulum system. However, it has been assumed that all the components of the state vector were available and no disturbances have been considered. This last assumption can be dangerous since during the design of the experimental facilities, we have been asked what is the targeted precision on the angular sensors. Simulations have been conducted in order to investigate this issue and we discovered that small offsets on these sensors systematically lead to a drift on the cart position ending with the cart hitting the mechanical stops. Such a simulation can be observed on Figure 5.17. This can be explained by the fact that the control law tries to compensate for a fictitious error that is due to offsets on the angular position measurements.

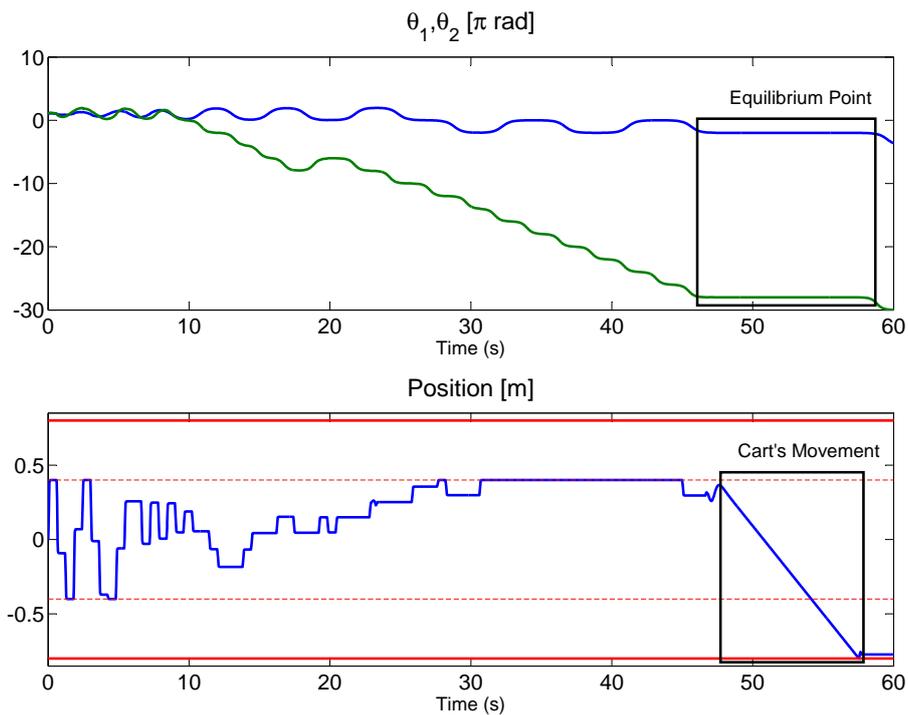


Figure 5.17: Closed-loop system system behavior with a bias of 1 *deg* on both angular sensors. The control law tries to compensate for fictitious errors on the angular position leading to a progressive drift on the cart position that ends with the cart hitting the mechanical stop.

This problem clearly underlines the need of a dedicated estimation scheme that re-

covers both the values of the unmeasured components of the state vector and such persistent offset errors on the measurement of the angular positions. Initially, a simple linear observer was tested that is fired at the instant where the control switches from the nonlinear to the linear law near the upward vertical position, but the error due to the initial state estimation at the switching instant destabilized systematically the system leading to the switch back from linear to nonlinear controller. This resulted in a system that can never be stabilized.

For this reason, a unified nonlinear observer that can turn on all the time regardless the active phase (nonlinear or linear controller) has been developed [Murilo and Alamir, 2007]. This observer design is explained in the following section.

5.5.2 Nonlinear Observer Design

Assume that only the angular and cart's position are measured, namely θ_1 , θ_2 and r . Let us also consider that the measurements of the angular positions θ_i of the pendulums are affected by constant offsets ε_1 and ε_2 , the output measurement vector y can be written as follows:

$$y = \begin{pmatrix} \theta_1 + \varepsilon_1 \\ \theta_2 + \varepsilon_2 \\ r \end{pmatrix} \quad (5.39)$$

Moreover, it is also assumed that the offset errors are upper bounded, namely there is some $\varepsilon_{max} > 0$ such that:

$$|\varepsilon_i| \leq \varepsilon_{max} \quad ; \quad i \in \{1, 2\} \quad (5.40)$$

Based on the above notation, a new extended state vector x_{ext} can be defined by:

$$x_{ext} = (r \quad \dot{r} \quad \theta_1 \quad \dot{\theta}_1 \quad \theta_2 \quad \dot{\theta}_2 \quad \varepsilon_1 \quad \varepsilon_2)^T \quad (5.41)$$

which leads to the following extended system, assuming that ε_1 and ε_2 have a constant behavior:

$$\ddot{r} = v \quad (5.42)$$

$$\ddot{\theta}_i = -\alpha_i(\cos \theta_i)v + \beta_i \sin \theta_i \quad i \in \{1, 2\} \quad (5.43)$$

$$\dot{\varepsilon}_i = 0 \quad i \in \{1, 2\} \quad (5.44)$$

If the term $\beta\theta_i$ is added to and retrieved from (5.43), a compact form can be obtained as follows:

$$\dot{x}_{ext} = Ax_{ext} + \phi(y, \varepsilon, v) \quad (5.45)$$

$$y = Cx_{ext} \quad (5.46)$$

where one can easily check that the pair (A, C) is observable while the nonlinear term $\phi(y, \varepsilon, v)$ is given by :

$$\phi(y, \varepsilon, v) = \begin{pmatrix} 0 \\ v \\ 0 \\ -\beta_1(y_1 - \varepsilon_1) + \beta_1 \sin(y_1 - \varepsilon_1) - \alpha_1 v \cos(y_1 - \varepsilon_1) \\ 0 \\ -\beta_2(y_2 - \varepsilon_2) + \beta_2 \sin(y_2 - \varepsilon_2) - \alpha_2 v \cos(y_2 - \varepsilon_2) \\ 0 \\ 0 \end{pmatrix}$$

This suggests to use the following structure for the observer:

$$\dot{\hat{x}}_{ext} = A\hat{x}_{ext} + \phi(y, S(\hat{\varepsilon}), v) - K_{obs}(\hat{y} - y) \quad (5.47)$$

in which K_{obs} is an observer gain that can be computed for the observable pair (A, C) using classical observer design methods (pole placement, optimal Kalman filter theory, etc.) and $S(\hat{\varepsilon})$ is a saturation map $S : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ defined as follows:

$$S_i(\hat{\varepsilon}) = \begin{cases} -\varepsilon_{max} & \text{if } \hat{\varepsilon}_i \leq -\varepsilon_{max} \\ +\varepsilon_{max} & \text{if } \hat{\varepsilon}_i \geq +\varepsilon_{max} \\ \hat{\varepsilon}_i & \text{otherwise} \end{cases}$$

It is worth noting that the output of the observer is not saturated, only $\hat{\varepsilon}$ in the nonlinear term $\phi(y, S(\hat{\varepsilon}), v)$ is saturated. This remark is quite important to understand the evolution of $\hat{\varepsilon}$ in the simulation results where it can be noticed that the estimated ε can temporarily exceed ε_{max} . The use of the saturation in the definition of the observer dynamic amounts to inject in the observer design the specific knowledge on the realistic offset that can affect the angular positions sensors.

In other words, the observer dynamic is built up by copying a saturated version of the extended system's dynamic together with a correction term that is based on the observability of the linear part. It is clear that if no offset error is considered, the convergence of the observer error is guaranteed since only the measured quantities y and v affect the nonlinear term ϕ . However, when unknown variables ε_1 and ε_2 disturb the output, the stability of the resulting observation scheme must be analyzed in details. This is discussed in the next section.

5.5.3 Convergence Analysis

In order to study the convergence of the observer, the dynamic of the estimation error $e = x - \hat{x}$ must be analyzed. From (5.45) and (5.47), the following expression is obtained:

$$\dot{e} = A_{obs}e + \phi_{nl}(y, \varepsilon, \hat{\varepsilon}, v) \quad (5.48)$$

where $A_{obs} = A - K_{obs}C$ is assumed to satisfy the following Lyapunov equation:

$$A_{obs}^T P + P A_{obs} = -I \quad ; \quad P > 0 \quad (5.49)$$

by choosing an appropriate gain matrix K_{obs} . Then, the nonlinear term ϕ_{nl} is given by:

$$\phi_{nl}(y, \varepsilon, \hat{\varepsilon}, v) := \phi(y, \varepsilon, v) - \phi(y, S(\hat{\varepsilon}), v) \quad (5.50)$$

We shall prove the following lemma:

Lemma 5.1. *There is a positive real $\mu > 0$ and a positive continuous map $\psi(\theta_1, \theta_2) \leq \mu$ that vanishes at $(0, 0)$ such that*

$$\|\phi_{nl}(y, \varepsilon, \hat{\varepsilon}, v)\| \leq \psi(\theta_1, \theta_2) \cdot \|\varepsilon - S(\hat{\varepsilon})\| \quad (5.51)$$

$$\leq 2\varepsilon_{max} \cdot \psi(\theta_1, \theta_2) \leq 2\varepsilon_{max} \cdot \mu \quad (5.52)$$

Proof: Remember that $y = \theta + \varepsilon$, the expression ϕ can be rewritten as follows:

$$\phi(y, \varepsilon, v) = \phi(\theta, 0, v)$$

Then, inequality (5.51) is a direct consequence of:

$$\|\phi(y, \varepsilon, v) - \phi(y, \hat{\varepsilon}, v)\| \leq \sup_{\substack{\theta \in [-\pi, +\pi]^2 \\ v \in [-v_{max}, v_{max}]}} \left\| \frac{\partial \phi}{\partial \theta}(\theta, 0, v) \right\| \cdot \|\varepsilon - S(\hat{\varepsilon})\|$$

since by definition of ϕ , one has

$$\sup_{\theta, v} \left\| \frac{\partial \phi}{\partial \theta}(\theta, 0, v) \right\| \leq \psi(\theta_1, \theta_2) = \left\| \begin{pmatrix} \beta_1(1 - \cos \theta_1) + \alpha_1 \cdot v_{max} \cdot \sin \theta_1 \\ \beta_2(1 - \cos \theta_2) + \alpha_2 \cdot v_{max} \cdot \sin \theta_2 \end{pmatrix} \right\|$$

which clearly satisfies the requirements of the lemma. As for (5.52), it is a direct consequence of assumption (5.40).

Therefore, based on lemma 5.1, the following proposition can be obtained:

Proposition 5.1. *The following statements hold for the dynamic of the estimation error e :*

1. *The evolution of e is globally bounded and tends to a final set that collapses to 0 with ε_{max} .*
2. *If the system parameters and the observer design are such that $2\mu\bar{\lambda}(P) < 1$ then the estimation error e globally asymptotically converges to 0 regardless the maximal value of the bias ε_{max} .*
3. *There is a sufficiently small $\theta_{max} > 0$ such that all closed-loop evolutions that keep $(\theta_1, \theta_2) \in [-\theta_{max}, +\theta_{max}]^2$ lead to a corresponding estimation error that asymptotically converges to 0.*

Proof:

Proof of (1) Consider the candidate Lyapunov function $V = e^T P e$ where P is the Lyapunov matrix defined in (5.49) and $\bar{\lambda}(P)$ its highest eigenvalue. If the derivative of V is computed together with (5.48), the following expression is obtained:

$$\dot{V} = -e^T e + \phi_{nl}^T P e + e^T P \phi_{nl}$$

which, together with lemma 5.1, clearly gives:

$$\dot{V} \leq -\|e\|^2 + 2\bar{\lambda}(P) \cdot \psi(\theta_1, \theta_2) \cdot \|\varepsilon - S(\hat{\varepsilon})\| \quad (5.53)$$

$$\leq -\|e\|^2 + 2\mu\bar{\lambda}(P) \cdot \|\varepsilon - S(\hat{\varepsilon})\| \quad (5.54)$$

Since the offset error is upper bounded according to (5.40), one has by definition of the saturation map:

$$\|\varepsilon - S(\hat{\varepsilon})\| \leq 2\varepsilon_{max} \quad (5.55)$$

Moreover, injecting the above inequality in (5.54) leads to the inequality:

$$\dot{V} \leq -\|e\|^2 + 4\mu\bar{\lambda}(P)\varepsilon_{max} \quad (5.56)$$

which clearly proves the first claim of the proposition.

Proof of (2) Using the inequality $\|\varepsilon - S(\hat{\varepsilon})\| \leq \|e\|$ in (5.53) gives:

$$\dot{V} \leq \left[-1 + 2\mu\bar{\lambda}(P) \right] \cdot \|e\|^2 \quad (5.57)$$

which clearly ends the proof of point (2).

Proof of (3) As a matter of fact, if the inequality $\psi(\theta_1, \theta_2) \leq \mu$ is not used in the derivation of (5.57), the following less pessimistic inequality may be obtained

$$\dot{V} \leq \left[-1 + 2\psi(\theta_1, \theta_2)\bar{\lambda}(P) \right] \cdot \|e\|^2$$

and since ψ is continuous and vanishes at $(0, 0)$, it comes that there is a sufficiently small θ_{max} such that

$$\max_{(\theta_1, \theta_2) \in [-\theta_{max}, +\theta_{max}]^2} \left[-1 + 2\psi(\theta_1, \theta_2)\bar{\lambda}(P) \right] < 0$$

This clearly ends the proof of the proposition.

It is worth noting that the simulation results presented in the next section suggest that the third item of proposition 5.1 holds in many cases. Indeed, very often, one can observe that the convergence of the estimation error is enforced as soon as small values of the angular positions are reached.

5.5.4 Simulation Results

In this section, some simulation results are presented in order to show the efficiency of the proposed observer. The parameters used in the simulations are the same as those shown in table 5.2. Moreover, the case $t_r = T_p$ is used. The observer response time was set to 0.005 s to be applied in the discrete version of (5.47) and K_{obs} was obtained from the optimal Kalman filter method. Two scenarios are shown corresponding to constant and variable bias respectively.

Figure 5.18 shows the evolution of θ_1 , θ_2 , the cart's position r and the estimation of ε_1 and ε_2 , with $[\varepsilon_{max1}, \varepsilon_{max2}] = [6, 4]$ deg respectively. Note that the observer converges to the neighborhood of the right values of ε during the swing-up phase, and converges definitely when the linear controller is active, with no discontinuities in the switch between controllers, which confirms the third item of proposition 5.1.

In the second scenario shown in figure 5.19, the variable bias $\varepsilon_1 = 3 + 0.06\sin(t)$ deg and $\varepsilon_2 = 5 - 0.06\sin(t)$ deg are introduced. Note that the observer can correctly retrieve the sinusoidal profiles. On the other hand, the resulting cart's movement becomes also oscillatory. This shows that the system is quite sensible to variable disturbances affecting the measurements around the upward vertical position, even for low amplitude bias values. This kind of behavior is also observed on the experimental results (section 5.7), where several model uncertainties are not considered

and the system becomes naturally oscillatory around the equilibrium point. Note that this kind of sensitivity does not occur when a simple pendulum is considered as our experience showed using the single pendulum system. The related results are not reported here in order to concentrate on the main topic.

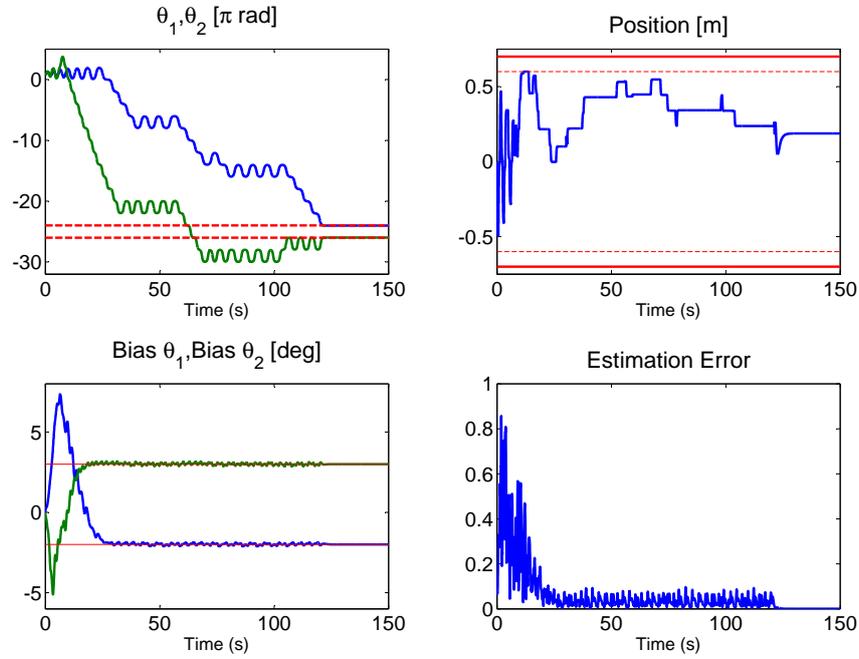


Figure 5.18: Evolution of θ_1 , θ_2 , r , the estimation of the offset error $[\varepsilon_1, \varepsilon_2] = [-2, 3]$ deg and the norm of the estimation error.

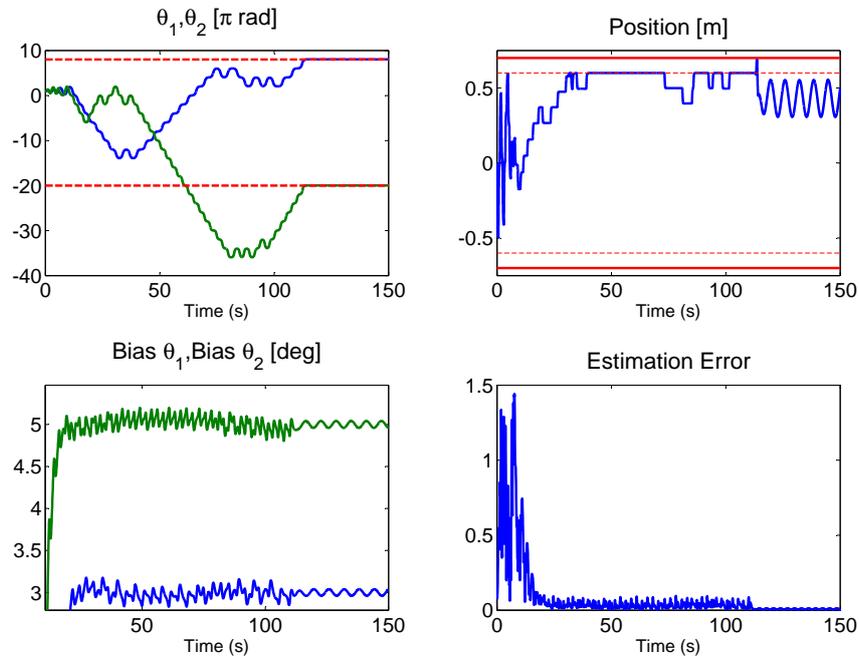


Figure 5.19: Evolution of the twin-pendulum system behavior under the presence of a variable bias $[\varepsilon_1, \varepsilon_2] = [3 + 0.06\sin(t), 2 = 5 - 0.06\sin(t)]$ deg on the angular sensors.

The final part of this chapter is dedicated to the implementation of the above presented control design to the experimental facility available at the Control Systems Department, Grenoble. First, some real-time implementation issues are discussed (identification, friction handling, etc.) before the experimental results are presented.

5.6 Real-Time Implementation

5.6.1 Identification Process

The identification of the physical parameters of the pendulums is done by performing open loop experiments with the cart being fixed in order to obtain the classical behavior as shown in figure 5.20.

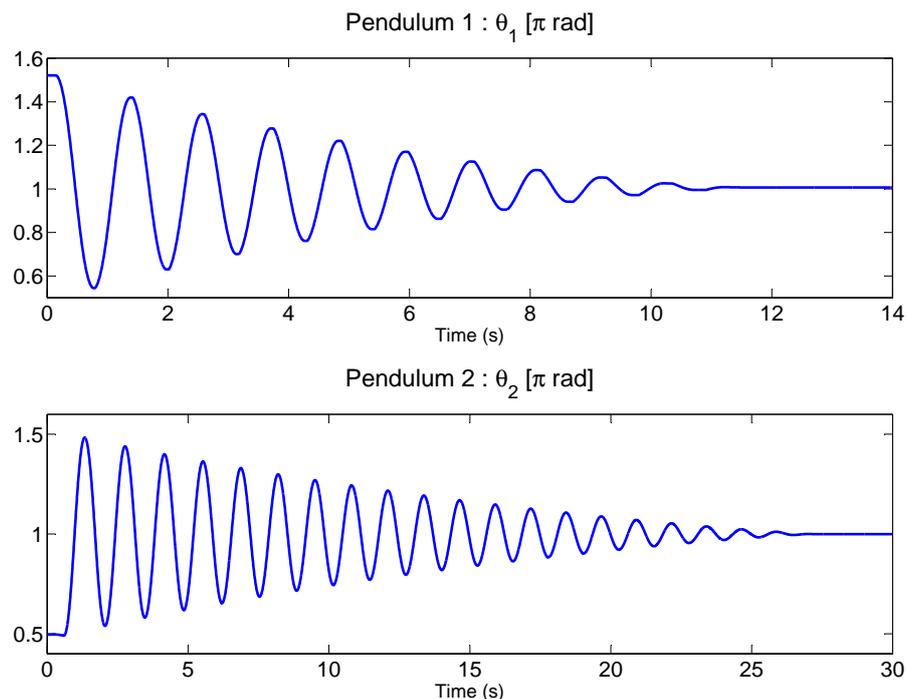


Figure 5.20: Open-loop experiments: the cart is fixed and the pendulums oscillate freely around the downward equilibrium.

In the previous scenarios, the simulation results did not consider the friction for the theoretical formulation. As it can be clearly seen from the open-loop identification experiments, friction is really present since the uncontrolled oscillations stop after

almost 10 periods for the first pendulum and around 20 periods for the second pendulum. Here, a typical representation using viscous and dry friction is introduced according to:

$$\ddot{\theta}_i = -\alpha_i v \cos \theta_i + \beta_i \sin \theta_i - k_{vi} \dot{\theta} - k_{ci} \text{sign}(\dot{\theta}) \quad ; \quad i \in \{1, 2\} \quad (5.58)$$

where k_{vi} and k_{ci} are the viscous and dry friction for pendulum i . When the cart is fixed, the equations become:

$$\ddot{\theta}_i = \beta_i \sin \theta_i - k_{vi} \dot{\theta} - k_{ci} \text{sign}(\dot{\theta}) \quad ; \quad i \in \{1, 2\} \quad (5.59)$$

It is worth noting that α_i and β_i are the physical parameters given by (5.9):

$$\alpha_i = \frac{m_i l_i}{m_i l_i^2 + I_i} \quad ; \quad \beta_i = g \alpha_i \quad ; \quad i \in \{1, 2\}$$

which means that α_i is known as soon as β_i is known. Using (5.59) together with the measured data, the parameters α_i , β_i and the friction coefficients can be identified. More specifically, identification has been done using the MATLAB optimization routine FMINCON. This function minimizes a constrained function of several variables. Figure 5.21 shows the resulting solution with the identified data.

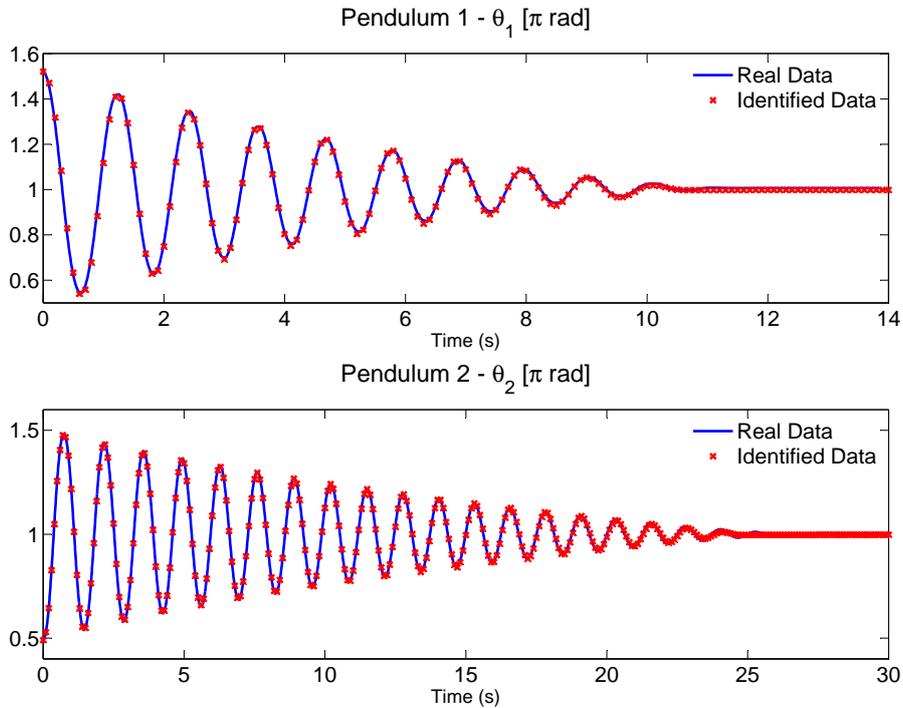


Figure 5.21: Real and identified data using the FMINCON routine.

Table 5.3 shows the values of the parameters α_i that can be obtained from (5.9) when injecting the measured weight and length of the pendulums.

Table 5.3: Parameters used in experiments

Pendulum	α measured	α identified	k_v	k_c
1	3.1856	3.2680	0.1537	0.3569
2	3.5129	3.5084	0.2969	0.8579

5.6.2 Dealing with Friction

In the derivation of the NMPC formulation presented in section 5.4, friction has not been explicitly considered. However, the previous identification experiment clearly shows that under 0 acceleration ($v = 0$), the pendulums do not keep the same orbit due to the presence of the high friction.

Nevertheless, as far as the NMPC design is concerned, the presence of friction does not change the basic idea of the strategy. However, the friction restricts the domain of control law parameters (sampling period, response time of the cart pre-compensator, etc.) under which successful closed-loop experiment can be obtained. This issue is discussed in the following four items.

The Choice of the Prediction Horizon

Although we have obtained a good identification results for the parameters of the friction model for the given open-loop identification scenario, everybody knows that the modeling of friction is almost impossible and the prediction of the evolution of systems that involves friction is a risky task. That is the reason why long term predictions have to be avoided as far as possible. The implication of this is that the prediction horizon is worth to be taken as short as possible in order to deal with meaningful prediction of the system behavior. In our case, the choice $T_p = t_r$ where t_r is the settling time for the cart pre-compensation loop seemed therefore to be the best choice. Indeed, remember that the philosophy of the control design imposes the constraint $T_p \geq t_r$, the choice $T_p = t_r$ represents the shortest *admissible* value for T_p in accordance with the above discussion.

Note that another advantage of this choice lies in the fact that the computation of the cost function needs less time for shorter prediction horizons and this may enable to reduce the sampling period and hence increase the reactivity of the system. This increases the possibility for the system to compensate for the effect of friction on the achievement of the control goal.

Number of Function Evaluations

In the theoretical formulation, only three function evaluations are calculated, which is enough under free-friction assumption. However, for real-time implementation, more iterations can be performed, in fact, as much as possible within the sampling period for control.

More precisely, an enumeration-like optimization is still performed but rather than three options as before, the number of candidate future set points for the cart position is now equal to the number of nodes on a uniform grid of possible positions in the admissible interval $[r(k) - d_r(E), r(k) + d_r(E)]$ as shown on Figure 5.22. The number of possible set-points is exactly equal to the number of resulting function evaluation n_{fe} . This number is constrained according to the following inequality:

$$n_{fe} \times \tau_{comp} \leq \tau_s \quad (5.60)$$

where τ_{comp} is the maximum time that may be needed to simulate the system equation over the prediction horizon T_p while τ_s is the control updating period.

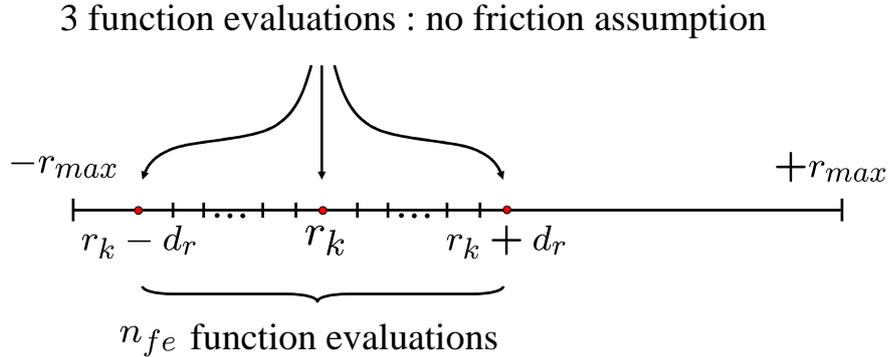


Figure 5.22: The computation of the optimal set-point r_d is done by checking the values of the cost function for n_{fe} uniformly distributed grid points in the interval $[r(k) - d_r(E), r(k) + d_r(E)]$. The number of candidate points and hence, the number of function evaluations depends on the available computation time according to the inequality constraint (5.60).

Hysteresis on the Switch

The previous issue deals with the friction problem during the swing-up phase. However, one of the conditions of switching (5.34) must also be adapted for real-time implementation. Remember that the condition defines a neighborhood around the equilibrium point where linear assumptions are valid.

$$\mathcal{A}_0 = \{z \in \mathbb{R}^5 | z^T S z \leq \rho_0\}$$

where $\rho_0 = z_0^T S z_0$ defines the boundary region and z_0 the limit state that can be defined as $z_0 = (0 \ \theta_0 \ 0 \ \theta_0 \ 0)^T$ which defines a region of attraction for the linear control law. However, this is a very sensitive and tight region and the switching action between controllers (NMPC to linear) may instantaneously lead the system outside of \mathcal{A}_0 . Then, a high-frequency switching may appear around z_0 , which is a quite undesirable phenomena for the system. Moreover, this problem appears very often in real-time implementation since model mismatches and friction introduces even more uncertainties to correctly define such a region. For this reason, a hysteresis in z is introduced in order to clearly define two boundaries as follows:

$$z_{in} = (0 \ 0 \ \theta_{in} \ 0 \ \theta_{in} \ 0) \quad (5.61)$$

$$z_{out} = (0 \ 0 \ \theta_{out} \ 0 \ \theta_{out} \ 0) \quad (5.62)$$

where θ_{in} is the angle that defines the boundary to go inside the region \mathcal{A}_0 and θ_{out} , to go outside the region. The z_{in} and z_{out} are the corresponding states at these angles. The condition to avoid this phenomena is to impose $\theta_{in} < \theta_{out}$, which clearly characterizes a hysteresis on the switch. Figure 5.23 illustrates the role of θ_{in} and θ_{out} to avoid the high-frequency switching phenomena.

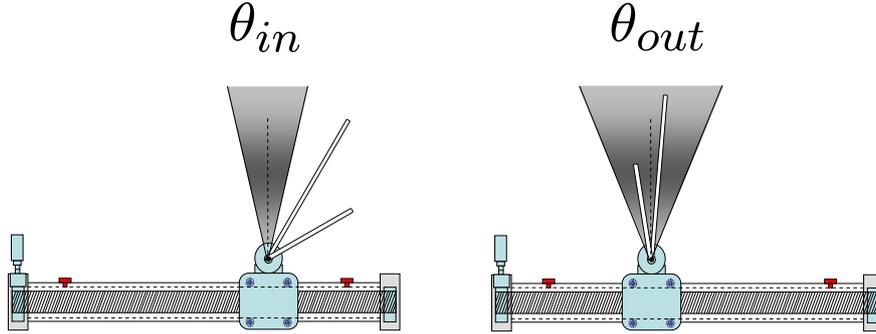


Figure 5.23: Schematic view of the hysteresis introduced for the switch condition.

The Final Position r^f

In all the above developments, the final position of the cart r^f was not imposed. In fact, unlike simulations, real-time experiments showed that adding a final target position on the cart during the final linearization based control leads to better performance in closed-loop. For this reason, a 6-th order system is considered for the linear phase. Consequently, the extended state vector z_e can be defined as

$$z_e = (r \ \dot{r} \ \theta_1 \ \dot{\theta}_1 \ \theta_2 \ \dot{\theta}_2) \in \mathbb{R}^6 \quad (5.63)$$

leading to the modified linear system

$$\dot{z}_e = A z_e + B v \quad (5.64)$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \beta_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \beta_2 & 0 \end{pmatrix} ; \quad B = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \alpha_1 \\ 0 \\ -\alpha_2 \end{pmatrix}$$

This results in a 6-dimensional gain vector K_L to be used in the expression $v(k) = -K_L z_e(k)$. Except for this detail, the remaining features of the strategy are kept identical to the one presented above.

With all these modifications at hand, some experimental results can be presented in the next section.

5.7 Experimental Results

The experimental validation was performed using the twin-pendulum platform of the Control Systems Department of the University of Grenoble. The real-time programs were developed in C, and used as S-functions under the MATLAB-SIMULINK environment and the XPC-TARGET was used as a real-time platform for the control implementation. We used a Pentium 4 running at 3 GHz. A User-Friendly Interface (GUI) was also developed in order to facilitate the set up of all the parameters concerning the NMPC controller, as shown in figure 5.24. All the technical details of this platform as well as the block diagrams used in experiments are shown in appendix C. Table 5.4 shows the main parameters used for the experimental validation. The sampling time τ_s was set to 5 ms.

Table 5.4: Parameters used in experiments

Parameter	Value	Parameter	Value
T_p	0.6 s	t_r	0.6 s
r_{max}^{RHC}	0.6 m	r_{max}	0.7 m
δ_r	0.4 m	n_{fe}	10
$[\alpha_1, \alpha_2]$	[2.59, 3.51] m^{-1}	$[\theta_{in}, \theta_{out}]$	[20, 30] deg
$[kc_1, kc_2]$	[0.3569, 0.8579]	$[kv_1, kv_2]$	[0.1537, 0.2969]

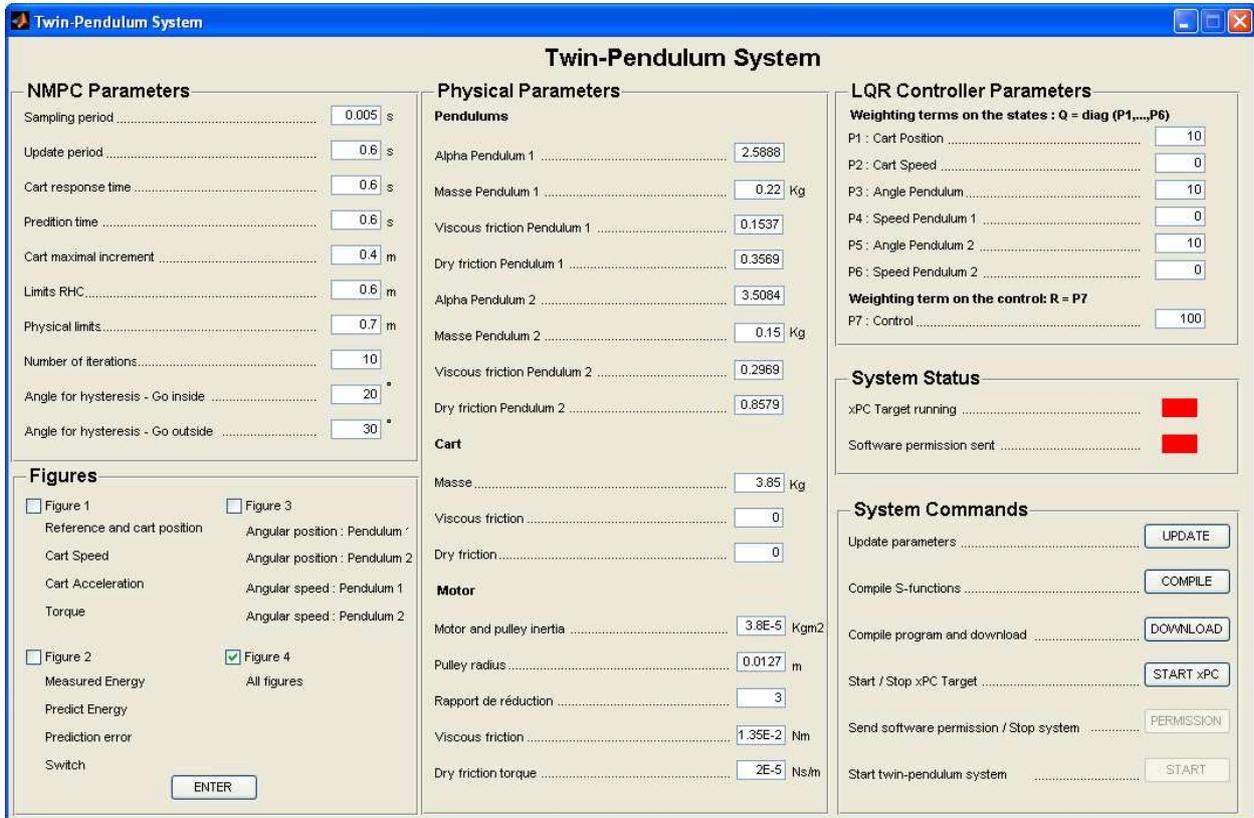


Figure 5.24: Graphical User Interface (GUI) for the twin-pendulum system.

The controller chosen for the linear phase was the Linear Quadratic Regulator (LQR), where Q and R are the weighting terms on the state and input, respectively. Here, the diagonal of Q was taken as $diag(Q) = (10 \ 0 \ 10 \ 0 \ 10 \ 0)$ (weighting on r , θ_1 , θ_2) and $R = 100$. The values of r , \dot{r} , θ_1 and θ_2 comes directly from the instruments and $\dot{\theta}_1$ and $\dot{\theta}_2$ are given by the filtered derivatives of their respective angular positions. Actually, the precision of the angular position sensors is very nice since each one is measured by an absolute encoder, which uses the code Gray 13 bits representing 8192 points (0.044 deg/point). This incites us to drop the use of the observer.

Figure 5.25 illustrates the first experimental scenario. The initial condition is $x_0 = (0 \ 0 \ \pi \ 0 \ \pi \ 0)^T$ which means that both pendulums are in the downward vertical at rest position, and the cart is at the origin with zero velocity. The controller is fired at $t = 5s$, before, the system runs in open-loop. Note that the constraints on the positions are correctly handled. Note also that due to the presence of model mismatches, the cart oscillates around the equilibrium point. Actually, this is mainly due to the friction on the cart, which is quite important and irregular along the

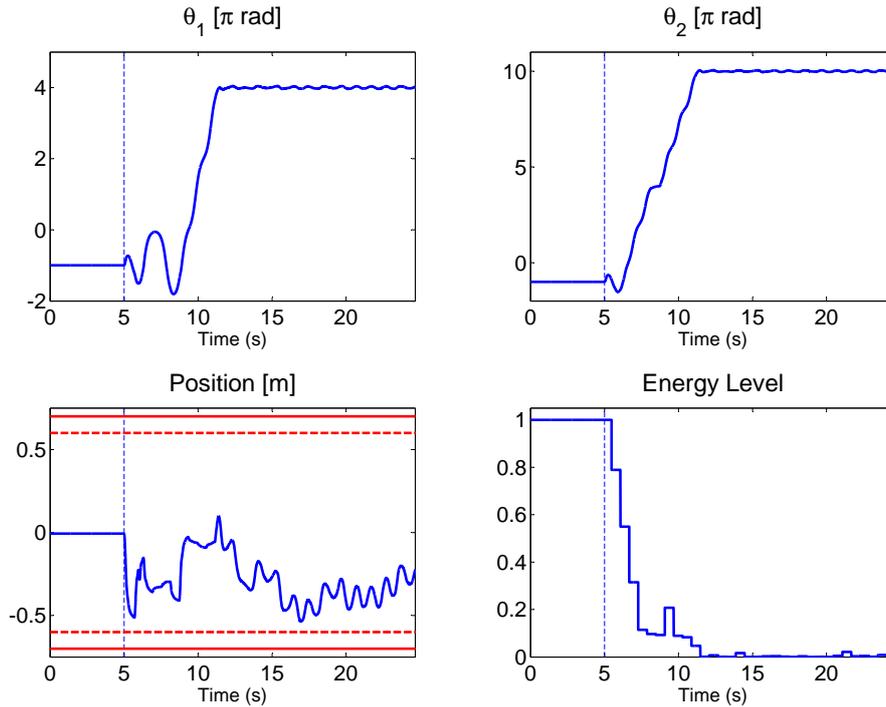


Figure 5.25: Experimental results showing the efficiency of the hybrid control strategy based on the parameterized contractive NMPC scheme in stabilizing the twin-pendulum system.

platform, with different dead-zones and nonlinearities. Some friction compensation techniques were tested with no significant improvement on the results [de Wit et al., 1995]. Anyway, regardless the model uncertainties, the contractive scheme of the NMPC manages to decrease the orbit level and the local controller stabilizes the system at the upward vertical at rest position.

The second scenario is shown in figure 5.26 starting from a different initial condition. Here, the initial angular speed of the pendulums are not equal to zero as in the previous case. Again, the control law is *fired* at $t = 5s$. Note that the constraints on the inputs has been aborted in order to avoid hitting the constraints at $r = -r_{max}$. Moreover, the virtual constraint r_{max}^{RHC} plays an important role to deal with the transient of the linear controller at $t \in [9, 15]s$. The orbit level then decreases to a neighborhood of zero and the system reaches the final manifold at almost $t \approx 20s$. Note here that due to friction, the decrease of the orbit level is not strictly monotonic but the cumulative effect of the transient decrease leads the system to the desired region around the targeted state.

The last scenario is illustrated on figure 5.27. The pendulums start at $\dot{\theta}_1 \neq 0$ and $\dot{\theta}_2 \neq 0$. Here, the objective is reached in less than 5 seconds. It is worth noting that the sampling period is only 5 *ms* and the computation time needed to perform iterations is around 3-4 *ms*, with $n_{fe} = 10$.

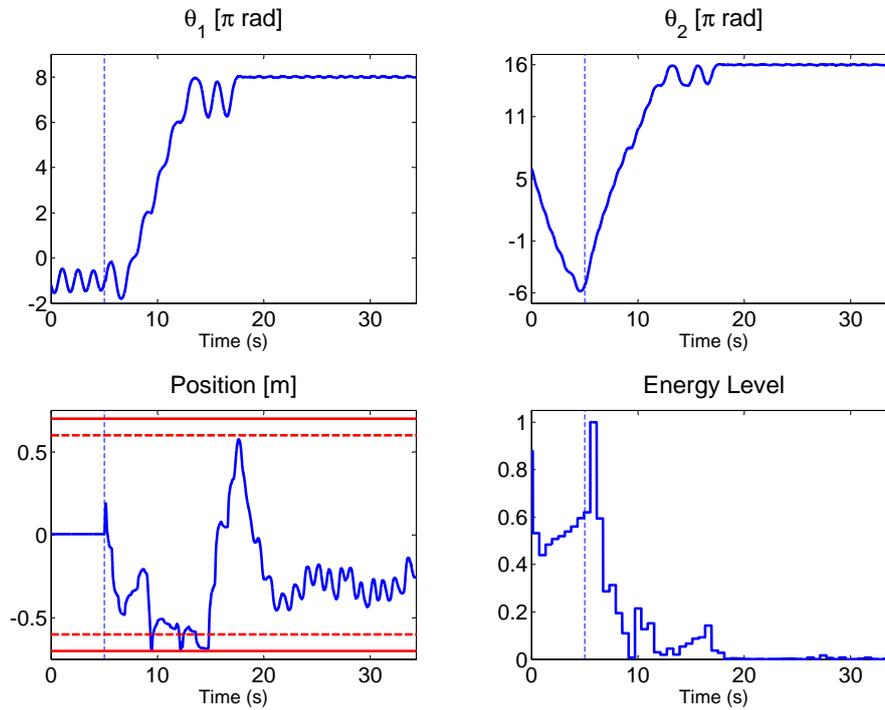


Figure 5.26: Second scenario starting from a different initial condition on the angular positions and speeds. Note the important role of the virtual constraint r_{max}^{RHC} at $t \in [9, 15]$ s.

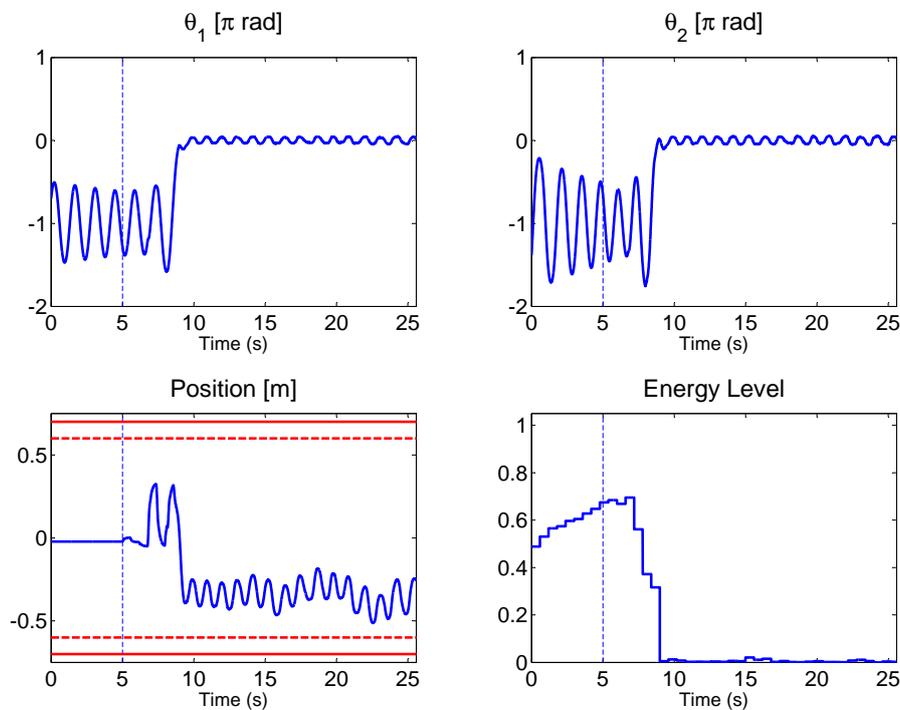


Figure 5.27: Last experiment showing the evolution of the twin-pendulum system under the parameterized NMPC scheme.

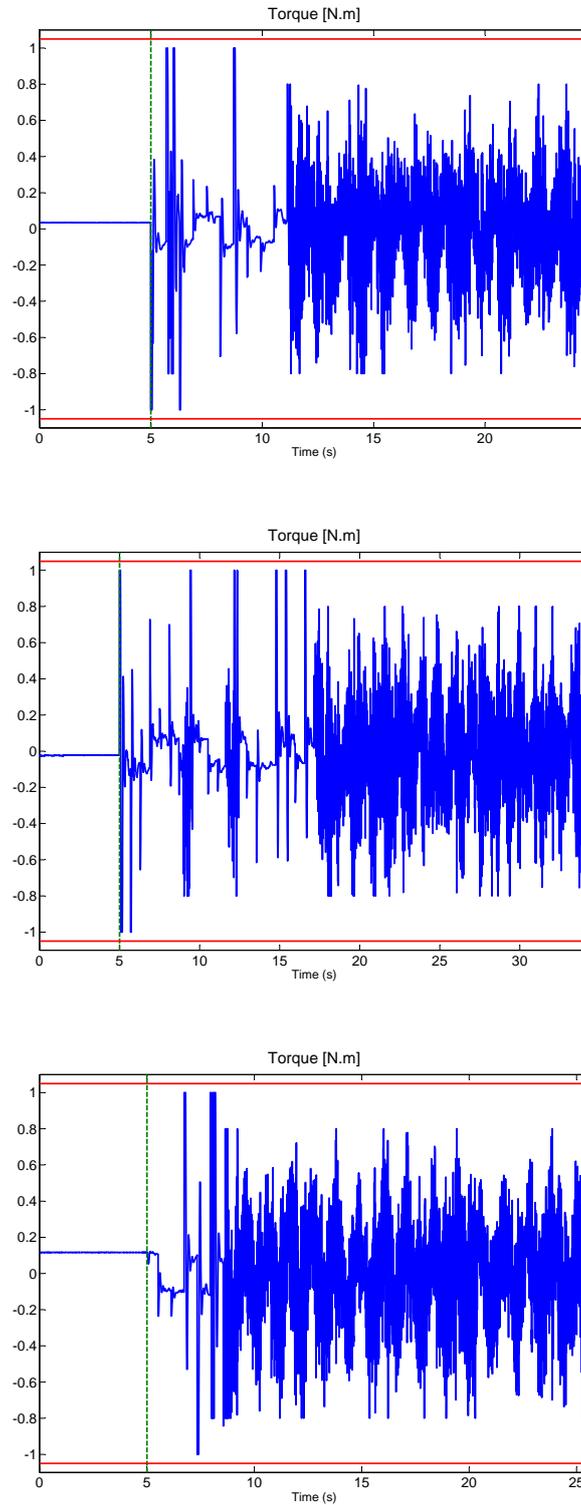


Figure 5.28: Evolution of the torque of the DC motor for the preceding scenarios.

Finally, figure 5.28 shows the behavior of the torque on the DC motor, which is the real system control input, for the three preceding scenarios. Note that the maximum displacement of the cart d_r was chosen in order to respect the maximum value of the torque, and hence this constraint is respected by construction, as shown in (5.22). Moreover, the torque does not converge to zero since the cart, in order to keep the pendulums at the upward vertical position, must deal with the model uncertainties and friction.

In order to better see how this system works, the videos of all the previous scenarios can be visualized in the web site [http : //www.lag.ensieg.inpg.fr/murilo/twinpendulum.html](http://www.lag.ensieg.inpg.fr/murilo/twinpendulum.html), which shows the experimental validation of the present control strategy.

5.8 Conclusion

This chapter presented a theoretical formulation and experimental validation of a parameterized NMPC approach for the twin-pendulum system. It was shown that the resulting control strategy deals with the main challenging tasks of this platform: the constraints on the state and inputs, nonlinearities and computation time. For the first one, a change of variables was proposed to obtain a new control input, which are the increments of the cart position. Then, a parameterized scheme based on such increments was used to minimize the energy level of the pendulums, resulting in a contractive-like approach. The proposed parametrization leads to a low-dimensional set of decision variables, since only three function evaluations must be computed at each sampling time. This reduces considerably the computation time, which guarantees real-time feasibility.

The experiments performed in the Control Systems Department of the University of Grenoble provided nice results and emphasized the efficiency of the parameterized approach. Despite of the presence of important friction and model uncertainties, the resulting control law was still able to decrease the energy level during the swing-up phase and stabilize the pendulums at the upward vertical position. It is worth noting that the computation time to perform optimization processes is always less than $5ms$ thanks to the well-posed parametrization structure. Although the fact that such dedicated structure characterizes a kind of problem dependency, the parameterized approach arises as being a powerful tool to deal with fast dynamic systems, such as mechatronic or robotic devices.

Conclusion and Future Work

Conclusion

In this PhD work, a parameterized NMPC strategy was used to deal with fast dynamic systems. For such systems, the recent idea of distributing the optimization problem during the system life-time motivated the development of several important contributions in this domain. Nevertheless, the approach presented in this dissertation holds some important advantages over the classical NMPC methods. The parameterized approach generally leads to a low-dimensional optimization problem while classical PWC control parametrization takes all components of control values (and in certain cases, even the state trajectory) as a degree of freedom, which may be prohibitive in a constrained computational context. In a sense, the parameterized approach adopts sub-optimality to tackle real-time requirements. Roughly speaking, solving exactly a simplified sub-optimal formulation may be a more interesting alternative than solving loosely the original one. On the other hand, the parameterized approach normally relies on the structure of the problem to be solved, and for instance, no general framework can be used to cover a wide-range of nonlinear applications.

The efficiency of the parameterized NMPC approach has been shown using two experimental platforms representing fast dynamics: the diesel engine air path and a twin-pendulum system.

The first real application is the diesel engine air path, where a multi-variable control based on the parameterized approach was proposed. It was shown that diesel engines are highly nonlinear and constrained systems. Therefore, the existing controllers for such systems generally use simplified models since the full nonlinear representation is quite complex to be used for standard control design. However, in this PhD work, it was shown that the parameterized NMPC solution developed for the diesel engine can be completely independent of the structure of the model being used. Therefore, more elaborated and reliable models can be taken into account by the NMPC controller. In addition, the resulting optimization problem is low-dimensional, reducing

drastically the computational burden. Simulations with the full nonlinear model provided quite nice results performing only 1 SQP iteration, which reinforces the idea of distributing the optimization problem during the system life-time.

Another important point to be highlighted is the fact that the generic feature of the proposed solution for diesel engines can be extended to any other system which may have a similar behavior. Here, three different nonlinear models were tested and all of them presented the same level of success.

The twin-pendulum system was the second application used for theoretical formulation and experimental validation of the parameterized NMPC scheme. Here, a change in the control variable was proposed in order to deal with the constraints on the inputs and states. Then, the increments on the cart position became not only the new control input, but also the parameter to be optimized for the NMPC strategy. As a result, this parametrization was used together with an energy-based method to formulate a contractive-like scheme. Again, the resulting optimization problem is quite simple to be solved which guarantees real-time feasibility. Experimental results performed on a real platform were quite positive and clearly show the efficiency of the proposed NMPC method.

The previous example shows that the parameterized NMPC approach may be an interesting solution for control design when mechatronic systems or similar devices are concerned. As a matter of fact, in most of the cases, it is possible to find a suitable and low-dimensional parametrization to cope with a particular structure of such systems.

Finally, some other important results of this PhD work are related to the development of two nonlinear observers: the first one, a MHO applied to the diesel engine, and the second one, for the twin-pendulum system. As far as the MHO is concerned, it is worth mentioned that one was successfully employed in a real application together with another predictive-like strategy, the parameterized NMPC scheme, resulting in a output feedback strategy. This is a quite consistent result, since one of the main drawbacks of MHOs is the real-time applicability. The second nonlinear observer designed for the twin-pendulum system also provided important results concerning the recover of a constant offset error. Moreover, it was also shown that the system may be quite sensitive when small variable uncertainties are added to the system. This system behavior can be clearly seen in real-time implementation.

Future Work

The solution developed for the diesel engine represents an important step to introduce some generic properties to the parameterized approach. Further works lies on the improvement of the proposed control scheme, in order to reduce the problem dependency, which is the main drawback of the method.

Concerning the diesel engine application, the next step of the proposed control strategy consists in using the full nonlinear model for experimental validation. The simulation results using such model were quite promising and the experiments will be hopefully performed once the diesel engine test bench is available.

The new SQP/Gradient/Trust Region algorithm presented in this dissertation has the important feature to be a generic gradient-free solver for optimization process. Therefore, the next step consists in developing an open source code and make it available for free download.

For the twin-pendulum system, as far as NMPC is concerned, there are no more significant theoretical formulations to be developed. A possible idea is to make the experimental twin-pendulum platform a benchmark in order to test and compare different kinds of control strategy.

Appendix A

Moving Horizon Observer Matrices

This appendix presents the computation of the matrices $\Phi(\bar{U}_k)$, $\Psi(\bar{U}_k)$, $\Omega(\bar{U}_k)$ and $\Gamma(\bar{U}_k)$, that are used for the moving horizon observer implemented to recover the states of the diesel engine model.

Given the following nonlinear system, depending on the inputs as follows:

$$\begin{cases} x^+ = [A(u, w)]x + [B(u, w)][u \quad w]^T \\ y = [C(u, w)]x \end{cases} \quad (\text{A.1})$$

And considering the following notations:

$$\begin{aligned} A_k &= A(u(k), w(k)) \\ A_{k-1} &= A(u(k-1), w(k-1)).A(u(k), w(k)) \\ &\vdots \\ A^{(-)} &= \prod_{i=1}^{N_o} A(u(k-N_o+i), w(k-N_o+i)) \end{aligned}$$

$$\begin{array}{lcl} B_k & = & B(u(k), w(k)) \\ B_{k-1} & = & B(u(k-1), w(k-1)) \\ \vdots & = & \vdots \\ B^{(-)} & = & B(u(k-N_o+1), w(k-N_o+1)) \end{array} \quad ; \quad \begin{array}{lcl} C_k & = & C(u(k), w(k)) \\ C_{k-1} & = & C(u(k-1), w(k-1)) \\ \vdots & = & \vdots \\ C^{(-)} & = & C(u(k-N_o+1), w(k-N_o+1)) \end{array}$$

Then, the matrices $\Phi(\bar{U}_k)$ and $\Psi(\bar{U}_k)$ can be obtained as follows:

$$\begin{aligned}\Phi(\bar{U}_k) &= A^{(-)} \\ \Psi(\bar{U}_k) &= [A^{(-)}.B^{(-)} \quad \dots \quad A_{k-1}.B_{k-1} \quad B_k]\end{aligned}$$

And the matrices $\Omega(\bar{U}_k)$ and $\Gamma(\bar{U}_k)$ can be calculated by proceeding straightforward computations :

$$\Omega(\bar{U}_k) = \begin{pmatrix} C_k \\ C_{k-1}A_{k-1} \\ C_{k-2}A_{k-2} \\ \vdots \\ C^{(-)}A^{(-)} \end{pmatrix}$$

$$\Gamma(\bar{U}_k) = \begin{pmatrix} 0_{2x2} & 0_{2x2} & 0_{2x2} & \dots & 0_{2x2} \\ C_k.B_k & 0_{2x2} & 0_{2x2} & \dots & 0_{2x2} \\ C_{k-1}.A_{k-1}.B_{k-1} & C_k.B_k & 0_{2x2} & \dots & 0_{2x2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C^{(-)}.A^{(-)}.B^{(-)} & \dots & \dots & C_k.B_k & 0_{2x2} \end{pmatrix}$$

Appendix B

Diesel Engine: Technical Data

The experiments were performed in the Institute for Design and Control of Mechatronical Systems of Department of Mechatronics at Johannes Kepler University of Linz. In this appendix, the technical data of the diesel engine test bench are shown. In order to respect the confidential terms, the test facilities are described according to [Wei, 2006], who worked in the same test bench, and only the block diagram used for simulations is presented.

B.1 Test Facilities

Diesel Engine

The M47D, 2.0 liter, diesel engine installed in this test cell is a product from BMW. It can generate torque 280Nm at 1750rpm and its output power at 4000rpm can reach 100KW. The air path system installs a Variable Geometry Turbocharger system and an Exhaust Gas Recirculation controlled by a pneumatic valve. The Electronic Control Unit is Type DDE3 from Bosch company.

Dynamometers

The dynamometer installed in this test cell is APA 100 from AVL which can generate maximum brake torque 285[Nm] and the highest speed can be 10000rpm. The function of the dynamometer is to simulate the vehicle load torque. It can act as an actuator to control the engine speed or some other control purposes. The diesel engine performance under different operation speed can be tested by controlling the dynamometer output torque.

Emission Equipment

MEXA-7000 from HORIBA company: It can be used for measuring raw and CVS-diluted exhaust gases from all vehicle and engine types for basic R & D, model certification, quality testing, and durability. It includes several analyzers in a single, compact cabinet to measure THC, CO, CO₂, O₂, NO/NO_x and HC over a wide dynamic range. The system is configurable for emissions from gasoline, diesel, or alternative fuel engines.

Puma System

The PUMA 2:0 system from AVL is totally independent which can implement many functions such as speed and torque control, vehicle simulations and so on. At the same time it can cooperate with other systems together to record some important data.

Autobox

In AutoBox all the dSPACE boards, which are listed in the following table, are embedded that are list in the following table

Table B.1: Board Description

Board	Function	Specification
DS1005	Processor Board	Power PC 750 Processor, 480Hz
DS4302	CAN Interface Board	4 independent CAN channels
DS4002	Timing and Digital I/O Board	8 timing , 32 digital, 2 external inputs
DS2201	Multi-I/O Board	20 A/D and 8 parallel D/A channels
DS2001	High Speed A/D Board	5 A/D channels

Matlab Real Time Workshop and dSPACE

The dSPACE can provide the dynamical monitoring and some manual operation functions such as to change some parameters, switch the control modal and so on. Matlab Real Time Workshop acts the function to compile the Simulink control file to some codes that can be run in the AutoBox hardware.

INCA Monitoring Software

INCA is a special software that can access the data and variables values in the ECU by CAN connection.

PUMA Monitoring and Control Software

PUMA system can provide some monitoring functions of diesel engine safety statues, record some variables and so on.

B.2 Diesel Engine Parameters

Table B.2: Full Nonlinear Model Parameters (4.3)-(4.5)

Variable	Value	Unit	Description
V_i	5.4	dm^3	Intake manifold volume
V_x	0.4	dm^3	Exhaust gas manifold volume
R	287.058	$J/kg/K$	Ideal gas constant
κ	1.4	-	Specific heat ratio
V_d	1.998	dm^3	Total engine displacement volume
c_p	1.0144	$kJ/kg/K$	Specific heat at constant pressure
p_a	101.3	kPa	Ambient pressure
τ_{EGR}	0.50	s	Constant time of the EGR valve
τ_{VGT}	0.21	s	Constant time of the VGT valve
η_m	0.7	-	Compressor efficiency

Table B.3: Empirical Model Parameters (4.6) and (4.7)

Variable	Value	Unit	Description - Central Values
y_c	[119.12,1183.60]	$[kg/h, hPa]$	Output [MAF,MAP]
w_c	[2248,19.144]	$[RPM, mg/cyl]$	Measurement disturbances [N,wf]
u_c	[30.02,47.94]	$[\%, \%]$	Input [EGR,VGT]

B.3 Block Diagram

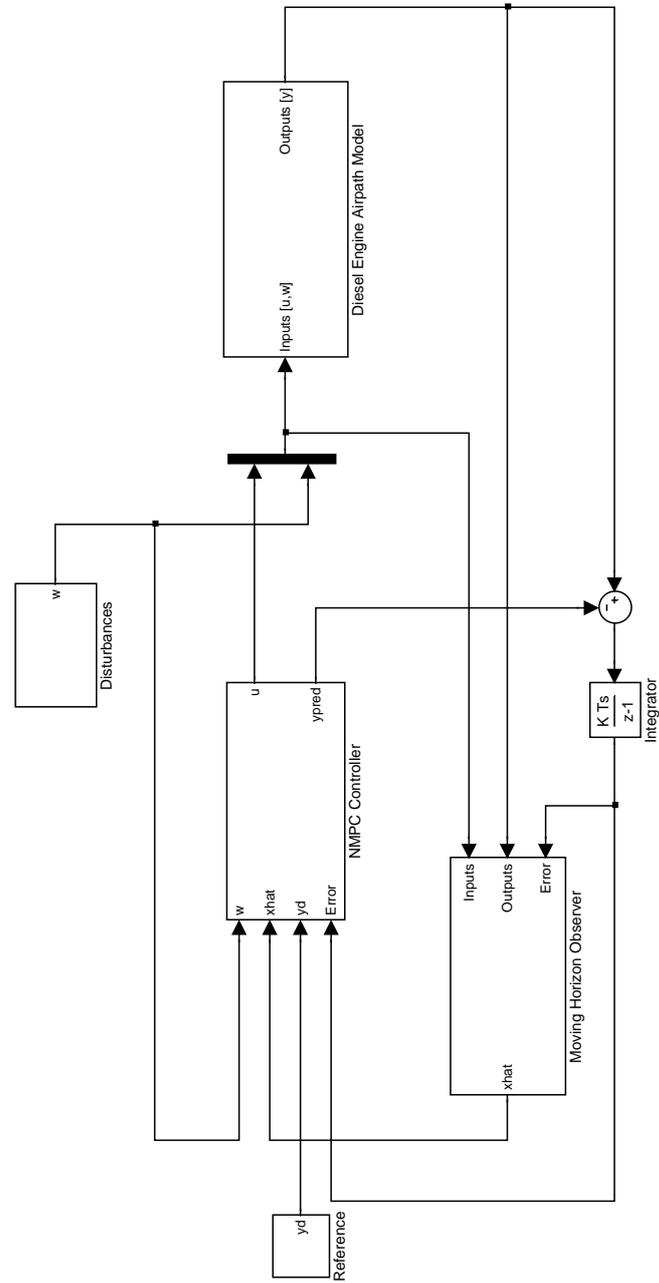


Figure B.1: Block diagram used for simulation. The moving horizon observer block is only necessary for models (4.6) and (4.7).

Appendix C

Twin-Pendulum System: Technical Data

The experiments were performed in the Control Systems Department of the University of Grenoble. This appendix presents some important technical data concerning the twin-pendulum platform.

C.1 Test Facilities

DC Motor

The motor used in the twin-pendulum platform is a DC Motor RS230C from Parvex. The parameters used for real-time implementation are shown in table C.2 and the complete description (in French) is presented in figure C.4.

Dynamo Tachometry

A Dynamo Tachometry (DT) is used to measure the cart speed. The model is a Parvex TBN203 with the constant $k_{DT} = 0.003V/RPM$. The block diagram used to convert the voltage input signal, is shown in figure C.1.

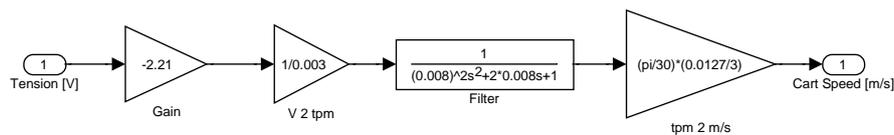


Figure C.1: Block diagram for the cart speed signal

Encoders

The encoders are used to measure the cart position r and the angles θ_1 and θ_2 .

For the cart position, an incremental encoder is used from Tekel, model TK 421, with 500 points/RPM.

For the angles, two absolute encoders are used with the code Gray 13 bits representing 8192 points. Then, the board PCI-DIO48H (table C.1) is used to acquire the digital signals that need to be converted. Figure C.2 shows the block diagram used to convert the 13-bit digital signal into the angular position between $-\pi$ and π .

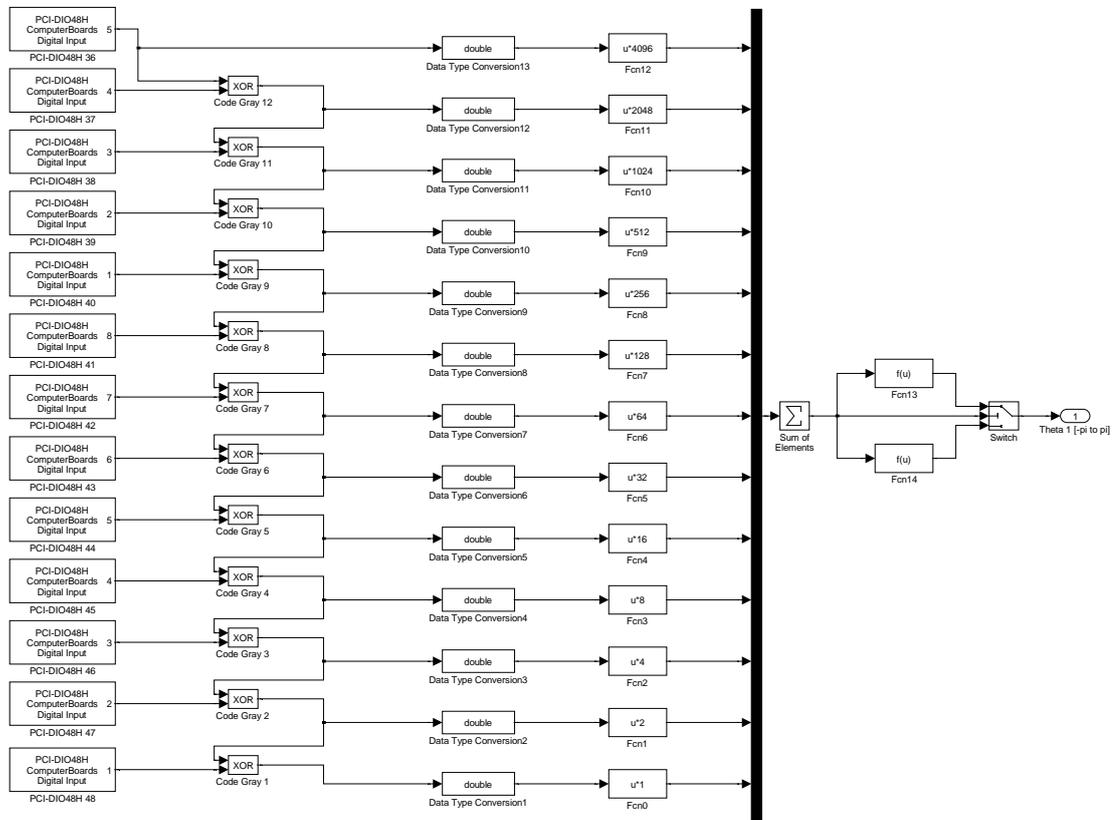


Figure C.2: Block diagram for the angular position signal

Interface Boards

The twin-pendulum platform uses three kinds of interface boards, all of them from Measurement Computing, which are listed in table C.1.

Table C.1: Interface Boards

Board	Function	Variables
PCI-DAS1200	Analog Input/Output	Cart Speed (\dot{r}), Torque (u)
PCI-DIO48H	Digital Input/Output	Angle 1(θ_1), Angle 2(θ_2)
PCI-QUAD04	Incremental Encoder	Position (r)

Anti-Aliasing Filter

The analog signal (cart speed) coming from the DT is filtered by a Butterworth 4th-order anti-aliasing filter. The board PCI-DAS1200 is used as shown in table C.1 and figure C.5 shows the anti-aliasing filter designed for the twin-pendulum platform.

Matlab Real Time Workshop and xPC Target

Contrarily to the diesel engine, the real-time environment used in this platform is the xPC Target. The software Matlab has been used for the development of the block diagrams and all the routines were written in C language. The system architecture is shown in figure C.3.

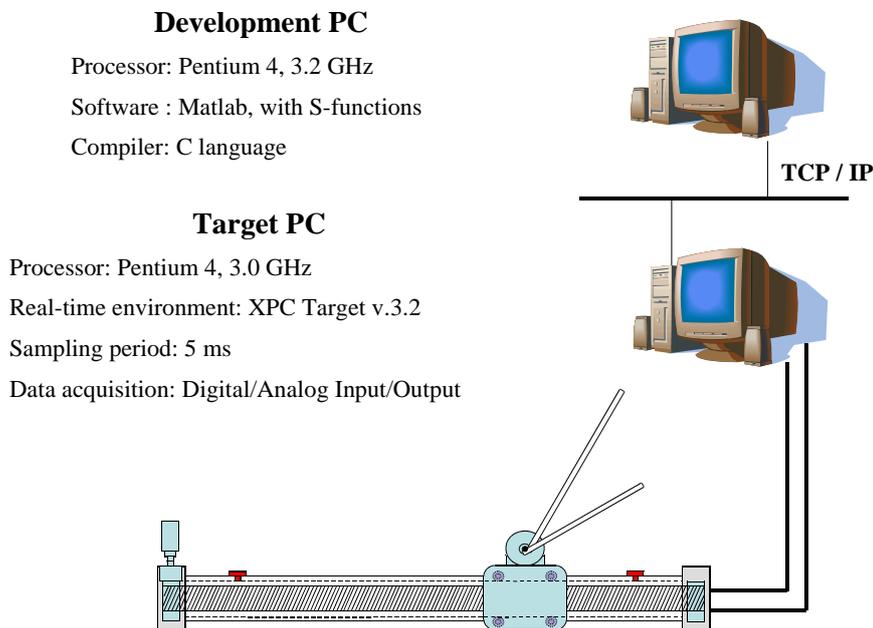
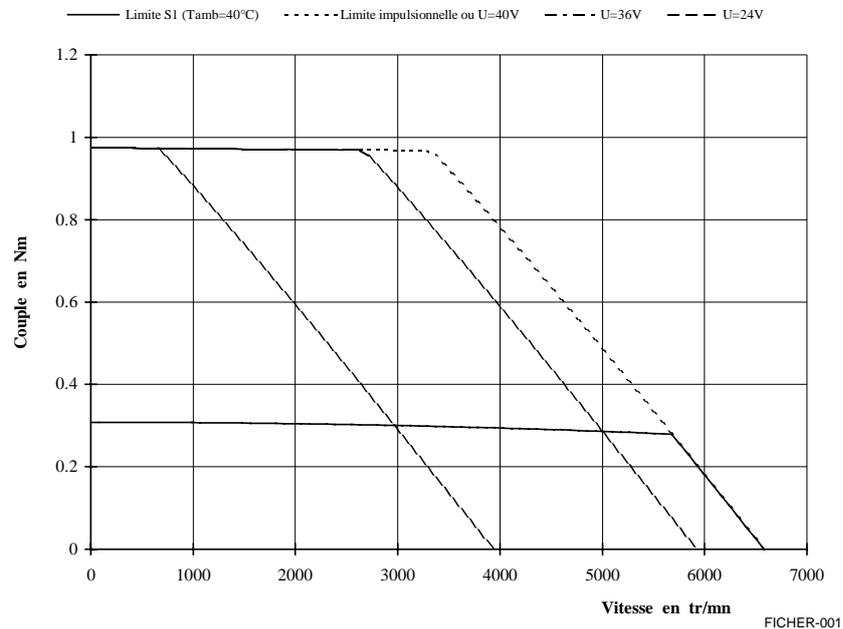


Figure C.3: System architecture of the twin-pendulum system.

<p>SERVOMOTEURS A COURANT CONTINU RS230C</p>	<p>PARVEX</p> <p>8 avenue du Lac BP249 F-21007 DIJON Cedex</p>
---------------------------------------------------------	-------------------------------------------------------------------------------

<i>Couple en rotation lente</i>	0.31	<i>N.m</i>	<i>M_o</i>
<i>Courant permanent rotation lente</i>	5.6	<i>A</i>	<i>I_o</i>
<i>Tension d'alimentation de définition</i>	24	<i>V</i>	<i>U</i>
<i>Vitesse de définition</i>	3000	<i>tr/mn</i>	<i>N</i>
<i>Tension maximale</i>	40	<i>V</i>	<i>U_{max}</i>
<i>Vitesse maximale</i>	6600	<i>tr/mn</i>	<i>N_{max}</i>
<i>Courant impulsionnel</i>	18	<i>A</i>	<i>I_{max}</i>
<i>Fem par 1000 tr/mn (25°C)</i>	6	<i>V</i>	<i>K_e</i>
<i>Coefficient de couple électromagnétique</i>	0.057	<i>N.m/A</i>	<i>K_t</i>
<i>Couple de frottement sec</i>	1.35	<i>N.cm</i>	<i>T_f</i>
<i>Coefficient de viscosité par 1000tr/mn</i>	0.2	<i>N.cm</i>	<i>K_d</i>
<i>Résistance du bobinage (25°C)</i>	0.67	<i>Ω</i>	<i>R_b</i>
<i>Inductance du bobinage</i>	0.42	<i>mH</i>	<i>L</i>
<i>Inertie rotor</i>	0.000026	<i>kg.m²</i>	<i>J</i>
<i>Constante de temps thermique</i>	7.5	<i>min</i>	<i>T_{th}</i>
<i>Masse moteur</i>	0.87	<i>kg</i>	<i>M</i>

Toutes les données sont en valeurs typiques pour des conditions d'utilisation standard



Création: 14 juil 1987	Edition: 25/Janv/1999	RS230C .f
------------------------	-----------------------	-----------

Figure C.4: Complete description of the DC motor

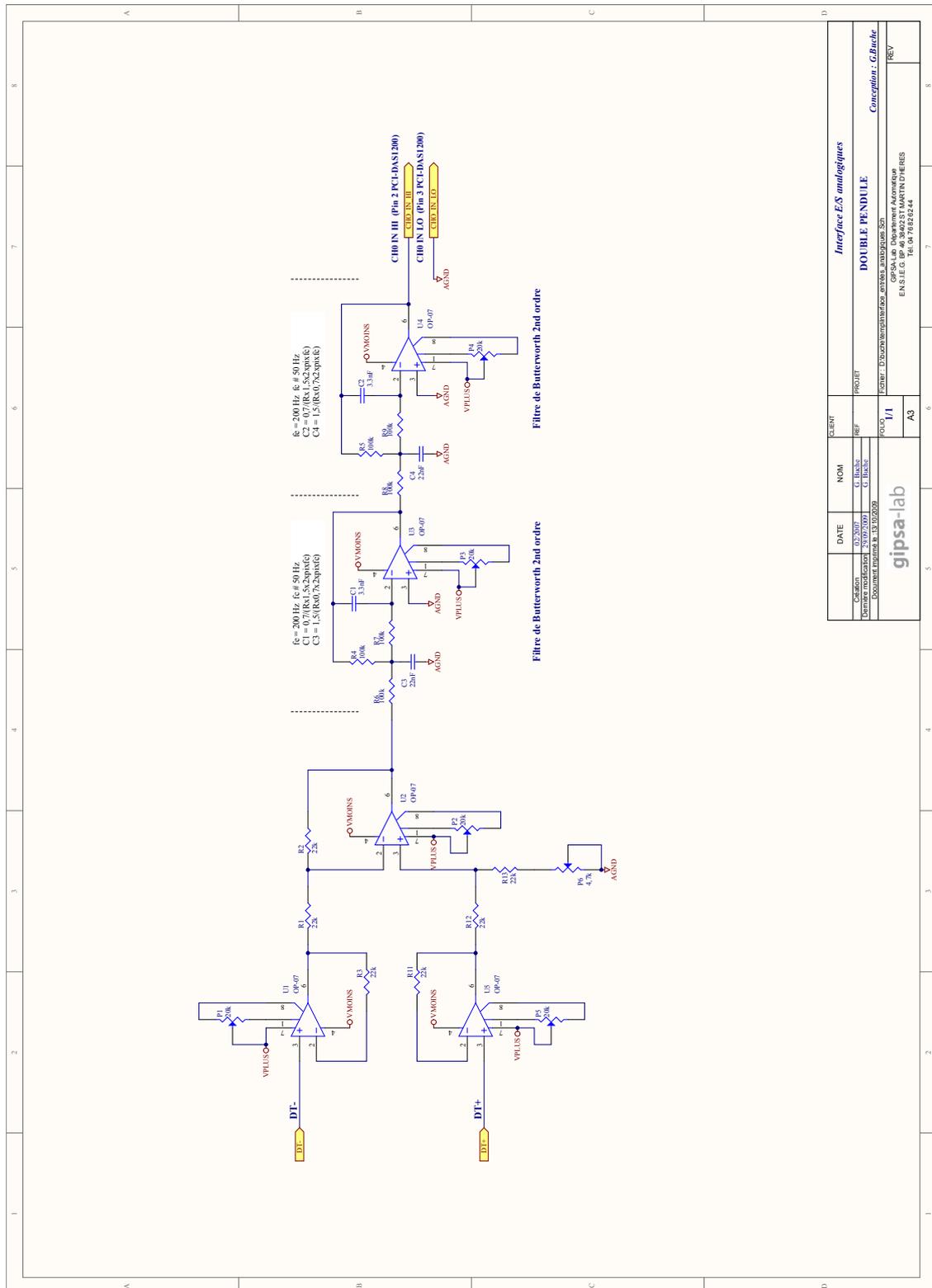


Figure C.5: Anti-aliasing filter used for the cart speed

C.2 Twin-Pendulum System Parameters

Table C.2: DC Motor Parameters

Variable	Value	Unit	Description
N_r	3	—	Reduction ratio
J_m	3.8e-5	$Kg.m^2$	Moment of inertia of the motor
J_a	0	$Kg.m^2$	Moments of inertia of the axis
k_{v_m}	2e-5	Ns/m	Motor viscous friction coefficient
k_{v_a}	0	Ns/m	Axis viscous friction coefficient
r_p	0.0127	m	Radius of the pulley
C_s	0.0135	Nm	Resistant dry torque

Table C.3: Physical Parameters: Pendulums and Cart

Variable	Value	Unit	Description
α_1	2.59	m^{-1}	Alpha of pendulum 1
m_1	0.22	Kg	Mass of pendulum 1
kv_1	0.1537	s^{-1}	Viscous friction of pendulum 1 (identified)
kc_1	0.3569	s^{-2}	Dry friction of pendulum 1 (identified)
α_2	3.5084	m^{-1}	Alpha of pendulum 2
m_2	0.15	Kg	Mass of pendulum 2
kv_2	0.2969	s^{-1}	Viscous friction of pendulum 2 (identified)
kc_2	0.8579	s^{-2}	Dry friction of pendulum 2 (identified)
m_0	3.85	Kg	Mass of the cart

C.3 Block Diagram

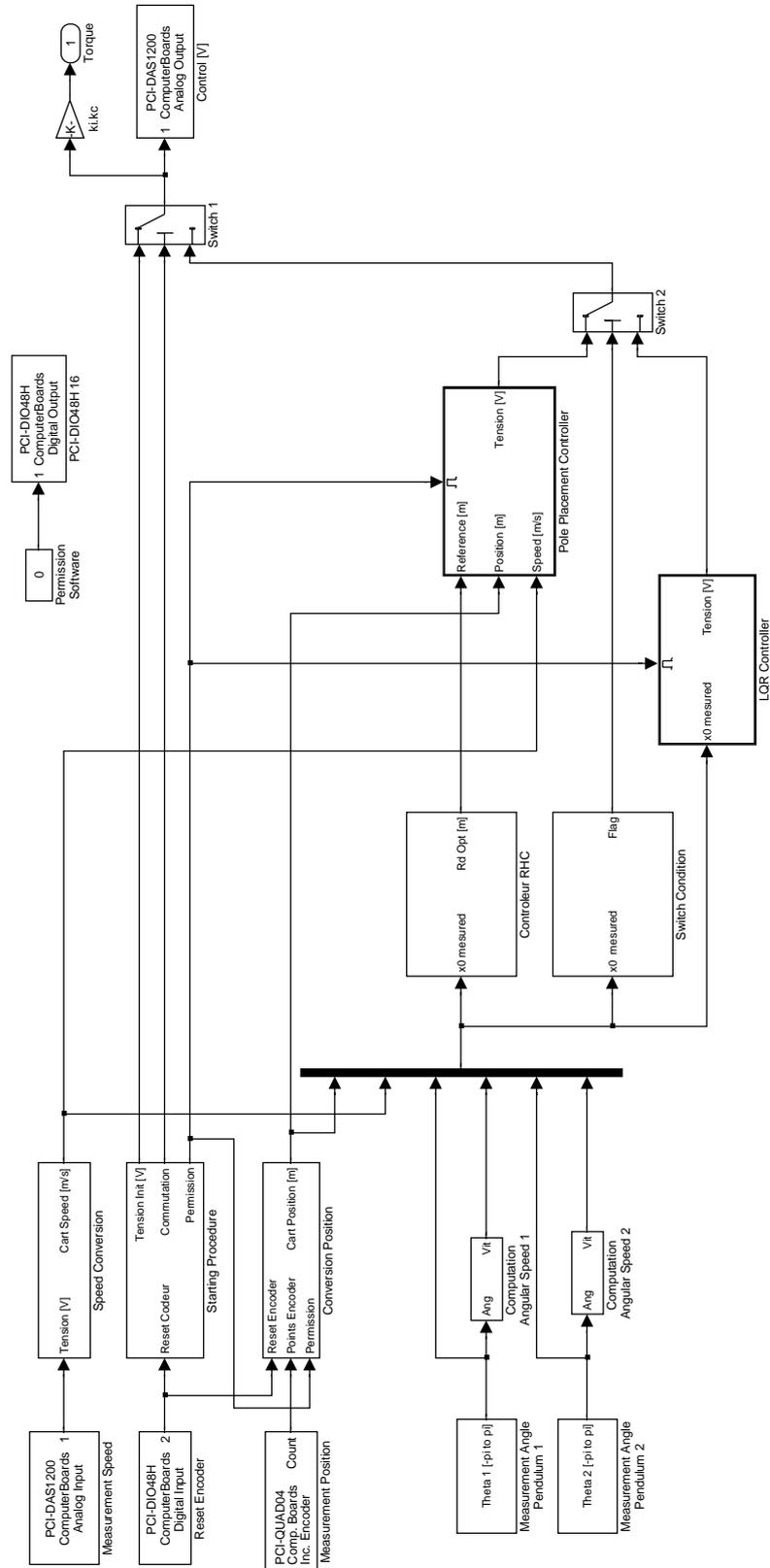


Figure C.6: Block diagram used in real-time implementation

Bibliography

- M. Alamir. Nonlinear receding horizon sub-optimal guidance law for the minimum interception time problem. *Control Engineering Practice*, 9(1):107 – 116, 2001.
- M. Alamir. *Stabilization of Nonlinear System Using Receding-Horizon Control Schemes: A parametrized approach for Fast Systems*. Lecture Notes in Control and Information Science. Springer, London, ISBN 1-84628-470-8, 2006a.
- M. Alamir. La commande prédictive non linéaire. In *La Commande Prédictive, Traité-IC2*, pages 157–195. Hermes-Lavoisier, 2006b.
- M. Alamir. A Low Dimensional Contractive NMPC Scheme for Nonlinear Systems Stabilization: Theoretical Framework and Numerical Investigation on Relatively Fast Systems. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *LNCIS*, pages 523–535. Springer, 2007.
- M. Alamir. *A Framework for Monitoring Control Updating Period in Real-Time NMPC*. Lecture Notes in Control and Information Science. Springer-Verlag, 2008.
- M. Alamir. Optimization based observers revisited. *International Journal of Control*, 72(13):1204–1217, 1999.
- M. Alamir and L. Calvillo-Corona. Further results on nonlinear receding horizon observers. *IEEE Transactions on Automatic Control*, 47(7):1188–1193, 2002.
- A. Alessio and A. Bemporad. A Survey on Explicit Model Predictive Control. *Assessment and Future Directions of Nonlinear Model Predictive Control*, 2008.
- R. Amari, M. Alamir, and P. Tona. Unified MPC strategy for idle-speed control, vehicle start-up and gearing applied to an automated manual transmission. In *Proceedings of the IFAC World Congress*, South Korea, 2008.
- R. Amari, M. Alamir, and P. Tona. Experimental evaluation of a hybrid MPC strategy for vehicle start-up with an automated manual transmission. In *Proceedings of the European Control Conference (ECC'09)*, Budapest, 2009.
- K. J. Åström and K. Furuta. Swing up a pendulum by energy control. *Automatica*, 36(2):287–295, 2000.
-

-
- R. Bellman. *Dynamic programming*. Princeton: Princeton University Press, 1957.
- A. Bemporad. Hybrid toolbox - User's guide, Univ. of Siena, Siena, Italy. <http://www.dii.unisi.it/hybrid/toolbox>, 2004.
- A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.
- M. J. Best. An algorithm for the solution of the parametric quadratic programming problem. In *Applied Mathematics and Parallel Computing*, pages 57–76. Physica-Verlag, 1996.
- L. T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering*, 8(3-4):243 – 247, 1984.
- L.T. Biegler and V.M. Zavala. Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization. *Computers & Chemical Engineering*, 33(3):575 – 582, 2009.
- H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 243–247, 1984.
- F. Borrelli, A. Bemporad, and M. Morari. Geometric Algorithm for Multiparametric Linear Programming. *Journal of Optimization Theory and Applications*, 108(3): 515–540, 2003.
- S. A. Bortoff. Robust swing-up control for a rotational double pendulum. In *Proceedings of the 13th IFAC World Congress*, pages 413–419, San Fransisco, California, 1996.
- A. Cervantes and L. T. Biegler. Optimization strategies for dynamic systems. In *Encyclopedia of Optimization*, pages 2847–2858. 2009.
- D. Chatterjee, A. Patra, and H. K. Joglekar. Swing-up and stabilization of a cart-pendulum system under restricted cart track length.
- J. Chauvin, G. Corde, and N. Petit. Constrained motion planning for the airpath of a Diesel HCCI engine. *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego*, 2006.
- J. Chauvin, G. Corde, N. Petit, and P. Rouchon. Motion planning for experimental airpath control of a diesel homogeneous charge-compression ignition engine. *Control Engineering Practice*, 16:1081–1091, 2008.
- H. Chen and F. Allgöwer. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica*, 34(10):1205 – 1217, 1998.
-

-
- U. Christen, K. J. Vantine, and N. Collings. Event-based mean-value modeling of Di Diesel engines for controller design. *SAE, Paper 2001-01-1242*, 2001.
- C.R. Cutler and B.L. Ramaker. Dynamic matrix control - a computer control algorithm. *Proceedings Joint Automatic Control Conference, San Francisco, CA*, 1980.
- C. Canudas de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3): 419 – 425, 1995.
- D. DeHaan and M. Guay. A Real-Time Framework for Model-Predictive Control of Continuous-Time Nonlinear Systems. *IEEE Transactions on Automatic Control*, 52(11):2047–2057, 2007.
- M. Diehl, H. G. Bock, and J. P. Schloöder. A Real-Time Iteration Scheme For Nonlinear Optimization in Optimal Feedback Control. *Siam Journal on Control and Optimization*, 43(5):1714–1736, 2005a.
- M. Diehl, R. Findeisen, F. Allgöwer, H.G. Bock, and J.P. Schlöder. Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Control Theory Appl.*, 152(3):296–308, 2005b.
- P. Dupraz. Modelisation et Commande Avancées d’un moteur diesel à injection directe. *Ph.D. dissertation, Institut National Polytechnique de Grenoble*, 1998.
- H. J. Ferreau. qpOASES Homepage. <http://homes.esat.kuleuven.be/~optec/software/qpOASES/>, 2007.
- H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy for fast parametric quadratic programming in MPC applications. *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, 2006.
- H. J. Ferreau, P. Ortner, P. Langthaler, L. del Re, and M. Diehl. Predictive control of a real-world Diesel engine using an extended online active set strategy. *Annual Reviews in Control*, 31:293–301, 2007.
- R. Findeisen. Model predictive control for fast nonlinear systems: Existing approaches, challenges and applications . In *Workshop at the 45th IEEE Conference on Decision and Control*, 2006.
- M. Fliess, J. Lévine, Ph. Martin, and P. Rouchon. On differentially flat nonlinear systems. *IFAC-Symp. NOLCOS’92, Bordeaux*, pages 408–412, 1992.
- P. F. Flynn, R. P. Durrett, G. L. Hunter, A. O. zur Loye, O. C. Akinyemi, J. E. Dec, and C. K. Westbrook. Diesel Combustion: An Integrated View Combining Laser Diagnostics, Chemical Kinetics, and Empirical Validation. *SAE, Paper 1999-01-0509*, 1999.
-

-
- C.E. Garcia and A.M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46:73–87, 1986.
- C. Gauthier. Commande Multivariable de la Pression d’Injection dans un Moteur Diesel Common Rail. *Ph.D. dissertation, Institut National Polytechnique de Grenoble*, 2007.
- K. Graichen, M. Treuer, and M. Zeitz. Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica*, 43(1):63 – 71, 2007.
- M. Herceg, T. Raff, R. Findeisen, and f. Allgower. Nonlinear model predictive control of a turbocharged diesel engine. In *Proceedings of the 2006 IEEE International Conference on Control Applications*, pages 2766–2771, 2006.
- J. B. Heywood. Internal combustion engines fundamentals. *McGraw-Hill*, 1988.
- T. Jacobs, D. Assanis, and Z. Filipi. The impact of exhaust gas recirculation on performance and emissions of a heavy-duty diesel engine. *SAE, Paper 2003-01-1068*, 2003.
- M. Jankovic and I. Kolmanovsky. Robust nonlinear controller for turbocharged Diesel engines. *Proceedings of the American Control Conference, Philadelphia*, 1998.
- T. V. Johnson. Diesel Emission Control in Review. *SAE, Paper 2001-01-0184*, 2001.
- M. Jung. Mean-value modelling and robust control of the airpath of a turbocharged Diesel engine. *Ph.D. dissertation, University of Cambridge*, 2003.
- R. E. Kalman. Contributions to the theory of optimal control. *Boletin Sociedad Matematica Mexicana*, pages 102–119, 1960.
- S. S. Keerthi and E. G. Gilbert. Optimal, infinite horizon feedback laws for general class of constrained discret time systems. Stability and moving horizon approximations. *Journal of optimization theory and applications*, 57:265–293, 1988.
- C. T. Kelley. *Iterative methods for linear and nonlinear equations*, volume 16. Frontiers in applied mathematics, 1995.
- P. Langthaler. Model Predictive Control of a Diesel Engine Airpath. *Ph.D. dissertation, Johannes Kepler University*, 2007.
- E. B. Lee and L. Markus. *Foundations of optimal control theory*. John Wiley and Sons, New York, 1967.
- D. B. Leineweber. The theory of MUSCOD in a nutshell. In *IWR-Preprint 96-19. Universität Heidelberg*, 1996.
-

-
- K. H. Lundberg and J. K. Roberge. Classical dual-inverted pendulum control. In *Proceedings of the 2003 IEEE Conference on Decision and Control*, pages 4399–4404, 2003.
- L. Magni and R. Scattolini. Robustness and robust design of MPC for nonlinear discrete-time systems. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *LNCIS*, pages 239–254. Springer, 2007.
- D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000.
- D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.
- H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006, 1995.
- M. Morari and S. L. D. Oliveira. Contractive model predictive control for constrained nonlinear systems. *Automatica*, 45(6):1053 – 1071, 2000.
- A. Murilo and M. Alamir. Output feedback design of a twin pendulum system in presence of sensor bias. In *Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems (NOLCOS'07)*, Pretoria, South Africa, 2007.
- A. Murilo, M. Alamir, and P. Ortner. Fast NMPC Scheme for a Diesel Engine Air Path. In *Proceedings of the European Control Conference (ECC'09)*, Budapest, 2009a.
- A. Murilo, M. Alamir, and P. Ortner. Experimental validation of a parameterized NMPC for a diesel engine. In *Proceedings of 2009 IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, Rueil Malmaison, France, 2009b.
- M. J. Van Nieuwstadt, P. E. Moraal, I. V. Kolmanovsky, A. Stefanopoulou, P. Wood, and M. Criddle. Decentralized and multivariable designs for EGR-VGT control of a Diesel engine. *IFAC Workshop on Advances in Automotive Control, Mohican State Park, OH*, 1998.
- K. Ogata. *Modern Control Engineering*. Prentice Hall, ISBN 0-13-060907-2, 2002.
- T. Ohtsuka. A Continuation/GMRES Method For Fast Computation of Nonlinear Receding-Horizon Control. *Automatica*, 40(4):563 – 574, 2004.
- P. Ortner and L. del Re. Predictive control of a Diesel engine air path. *IEEE Transactions on Control Systems Technology*, 15(3):449–456, 2007.
- P. Ortner, R. Bergmann, H. J. Ferreau, and L. del Re. Nonlinear Model Predictive Control of a Diesel Engine Airpath. *IFAC Workshop on Control Applications of Optimisation, Jyvaskyla*, 2009.
-

-
- G. Pannocchia, J. B. Rawlings, and S. J. Wright. The Partial Enumeration Method for Model Predictive Control: Algorithm and Examples. *Technical report 2006-01. Texas-Wisconsin Modeling and Control Consortium*, 2006.
- A. Plianos and R. Stobart. Dynamic feedback linearization of Diesel engines with intake variable valve actuation. *IEEE Transactions on Control Systems Technology*, pages 455–460, 2007.
- V. V. Pontryagin, Y. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers Inc. New York, 1962.
- S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.
- J. Richalet, A. Rault, J. L. Testud, and J. Papon. Algorithmic control of industrial processes. *Proceedings of the Fourth IFAC symposium on identification and system parameter estimation*, pages 1119–1167, 1976.
- J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control : Applications to industrial processes. *Automatica*, 14(5):413 – 428, 1978.
- Y. Shimizu, T. Ohtsuka, and M. Diehl. Multiple-Shooting-Based Algorithm with Continuation/GMRES Method for Nonlinear Receding Horizon Control. *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, 2006.
- P. Tøndel, T. A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3):489 – 497, 2003.
- M. J. van Nieuwstadt, I. V. Kolmanovsky, and P. E. Moraal. Coordinated EGR-VGT control for Diesel engines: An experimental comparison. *SAE, Paper 2000-01-0266*, 2000.
- A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.
- X. Wei. Advanced LPV techniques for Diesel engines. *Ph.D. dissertation, Johannes Kepler University*, 2006.
- X. Xin and M. Kaneda. Analysis of the energy-based control for swinging-up two pendulums. *IEEE Transactions on Automatic Control*, 50(5):679–684, 2005.
- E. Zafiriou. Robust model predictive control of processes with hard constraints. *Computers & Chemical Engineering*, 14(4-5):359 – 371, 1990.
-

-
- V. M. Zavala and L. T. Biegler. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1):86 – 93, 2009.
- V. M. Zavala, C. D. Laird, and L. T. Biegler. Fast solvers and rigorous models: can both be accommodated in NMPC? *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, 2006.
- W. Zhong and H. Röck. Energy and passivity based control of the double inverted pendulum on a cart. In *Proceedings of the 2001 IEEE International Conference on Control Applications*, pages 896–901, Mexico City, September 2001.
-

List of Publications

International Journal

M. Alamir and A. Murilo. Swing-up and stabilization of a Twin-Pendulum under state and control constraints by fast NMPC scheme. *Automatica*, vol 44, pp. 1319-1324. (2008).

Book Chapter

M. Alamir, A. Murilo, R. Amari, R. Fürhapter and P. Ortner (2009) On the use of parametrized NMPC in real-time automotive control. In *Automotive Model Predictive Control*, Luigi Del Re (Ed). Lecture Notes on Communication and Information Science.Springer-Verlag-Series.

International Conferences

A. Murilo and M. Alamir. Output feedback design of a twin pendulum system in presence of sensor bias. In *Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems (NOLCOS'07)*, Pretoria, South Africa, 2007.

A. Murilo, M. Alamir, and P. Ortner. Fast NMPC Scheme for a Diesel Engine Air Path. In *Proceedings of the European Control Conference (ECC'09)*, Budapest, 2009a.

A. Murilo, M. Alamir, and P. Ortner. Experimental validation of a parameterized NMPC for a diesel engine. In *Proceedings of 2009 IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, Rueil Malmaison, France, 2009b.



Grenoble Images Parole Signal Automatique

Grenoble INP

Contributions à la Commande Prédictive Non Linéaire pour les Systèmes à Dynamiques Rapides

Résumé:

Cette thèse adresse le problème de la commande prédictive non linéaire (CPNL) appliquée aux systèmes à dynamiques rapides. Pour ces systèmes, le temps de calcul disponible peut s'avérer insuffisant pour la résolution des problèmes d'optimisation fortement non linéaires et contraints. Dans ce contexte, l'approche paramétrique utilisée dans cette thèse peut s'avérer un choix pertinent pour résoudre ce type de problèmes. Deux exemples d'application sont présentés pour souligner les avantages de la méthode paramétrique: le circuit d'air d'un moteur diesel et le système des pendules jumeaux sur un chariot. Une validation expérimentale sur les deux procédés est aussi proposée pour montrer l'efficacité des solutions proposées.

Mots Clefs: Commande prédictive non linéaire, systèmes rapides, implémentation temps-réel, approche paramétrique, moteur diesel, pendules jumeaux, commande sous contraintes.

Contributions on Nonlinear Model Predictive Control for Fast Systems

Abstract:

This thesis addresses the challenging problem of applying nonlinear model predictive control (NMPC) to fast dynamical systems. For these systems, the available computational time may be insufficient to solve the underlying non convex optimization problems. In this context, the parameterized approach may be a potential candidate solution to deal with fast systems. Two practical applications are shown to emphasize the advantages of the parameterized approach: the diesel engine air path and the twin-pendulum system. Experimental validation results are proposed for the two processes in order to show the efficiency of the proposed framework.

Keywords: Nonlinear model predictive control, fast systems, real-time implementation, parameterized approach, diesel engine, twin-pendulum, constrained control.