KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT WERKTUIGKUNDE
AFDELING PRODUCTIETECHNIEKEN,
MACHINEBOUW EN AUTOMATISERING
Celestijnenlaan 300B, 3001 Heverlee (Leuven), Belgium

Time optimal control of mechatronic systems through embedded optimization

Jury:
Prof. dr. ir. H. Hens (voorzitter)
Prof. dr. ir. J. Swevers (promotor)
Prof. dr. M. Diehl (promotor)
Prof. dr. ir. J. De Schutter
Prof. dr. W. Saeys
Prof. dr. dipl-ing. R. Findeisen
Dr. ir. G. Pinte

Proefschrift voorgedragen
tot het behalen van de
graad van Doctor in de
Ingenieurswetenschappen

door

Lieboud Van den Broeck

2011D09

July 2011

© Katholieke Universiteit Leuven – Faculty of Engineering
Celestijnenlaan 300B, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2011/7515/89
ISBN 9789460183874

Dankwoord

Hier voor jullie ligt het resultaat van enkele jaren doctoreren of toch het zichtbare gedeelte ervan. Toen ik op mechanica begon, was mijn kennis van optimale controle van mechatronische systemen nagenoeg onbestaande. Ondertussen heb ik echter mogen ervaren dat het toch wel boeiend was om dieper op deze materie in te gaan. Doctoreren houdt meer in dan het eindresultaat alleen en ik kijk dan ook met een zeer positief gevoel terug op de voorbije jaren. Dit doctoraat had ik echter nooit kunnen maken zonder de steun en hulp van verschillende personen en deze zou ik dan ook graag danken.

Eerst en vooral gaat mijn dank uit naar mijn twee promotoren. Jullie vormden voor mij samen een ideaal promotorenpaar. Bedankt om mij tegelijkertijd te steunen waar nodig en mij toch ook veel vrijheid te geven qua onderzoek. Ook jullie combinatie om enerzijds telkens voor de grootste uitdagingen te willen gaan en anderzijds alle processen zeer goed te willen doorgronden was een zeer aangename en nuttige combinatie. Jan, bedankt voor alle hulp in verband met mechatronische en controle problemen maar vooral ook voor je grote bereikbaarheid en de hulp bij het schrijven van mijn papers. Ook de jaarlijkse controlegroepbarbecue apprecieerde ik zeker. Moritz, bedankt voor de ondersteuning bij de meer numerieke en optimalisatie problemen. Het FWO wil ik danken voor de financiële ondersteuning die dit doctoraat mogelijk maakte.

Daarnaast wil ik ook mijn juryleden bedanken voor het nalezen van de thesis en hun kritische commentaar hierop. *Joris, Greg, Wouter and Rolf, Thanks for the valuable comments on my text. They certainly have improved its quality.*

Doctoreren is over het algemeen een individueel gericht werk. Gelukkig doe je dit echter niet op een eiland. Daarom wil ik ook graag alle collega's bedanken. Hierbij denk ik in de eerste plaats aan alle collega's van de controlegroep. Verder wil ik ook alle collega's danken die meededen aan de niet-werk gerelateerde activiteiten zoals het kaarten, sporten, happy hour, ...Een speciaal woord van dank hierbij aan Bart, Elke en Pieter. Verder wil ik ook zeker de ATP collega's danken voor hun administratieve en technische hulp in de loop van mijn doctoraat.

Ook dank aan alle vrienden voor de ontspanningsmomenten in de loop van de week. Dankzij de spelletjesavonden, het sporten, terrasjes, . . . en ook wel dankzij de gedeelde doctoraatservaring van velen onder jullie had ik ruimschoots de mogelijkheid om regelmatig stoom af te blazen.

Ten laatste wil ik ook graag mijn familie bedanken voor hun steun. Bliede, bedankt om er te zijn als zus. Je helpt me om regelmatig eens alles in perspectief te plaatsen. Mijn ouders wil ik heel graag bedanken voor hun onvoorwaardelijke steun en liefde. Jullie zijn altijd ter beschikking als een luisterend oor en bieden me een plaats waar ik me altijd thuis kan voelen, bedankt.

Lieboud

Abstract

In all levels of society, a lot of effort is put in the optimal use of resources as e.g. energy, labor and time. Resource optimization is also important in the control of mechatronic systems. For these applications, typically either the required time or the energy consumption to perform an action is minimized. This thesis develops controllers which aim to minimize the time required to perform a point-to-point motion, i.e. the settling time of the system. These controllers are developed within the model predictive control framework (MPC). In this framework, the system input is determined by solving on-line, during every sampling period, an optimization problem. This on-line optimization allows to take systems constraints like actuator saturation, directly into account. The time-optimal controllers have been developed for mechatronic systems with sampling periods in the order of milliseconds. Hence, the fast solution of these problems is an important design parameter. The main contributions of this research are as follows. First, the minimization of settling time has been formulated as an optimization problem within the MPC framework. Then, the structure of these optimization problems has been analyzed and exploited such that these problems can now be formulated with enough variables to be applicable for relevant mechatronic applications while still being solvable within a few milliseconds. All developed time-optimal controllers have been validated experimentally on representative mechatronic systems as linear motors and an overhead crane. This experimental validation shows that with the developed controllers sampling periods of a few milliseconds are attainable and that the settling time can be reduced considerably in comparison with linear controllers and traditional MPC controllers.

Within this global framework of minimizing settling time, three controllers have been designed. First, a time-optimal feedforward controller has been developed. This controllers generates a reference trajectory which minimizes the settling time for point-to-point motions. This feedforward controller has been developed as a more performant alternative to linear prefilters. Then, a time-optimal feedback controller has been developed. The introduction of feedback allows to reject disturbances and to remove steady state errors. Last, a control scheme which combines the time-optimal controllers with linear feedback controllers, has been

proposed. This scheme allows to fulfill the benchmark requirements of an industrial linear motor, i.e. not only a small settling time but also an absolute settling accuracy in the submicrometer range.

Korte inhoud

In de huidige maatschappij staat het optimaal aanwenden van productiemiddelen zoals energie, arbeid en tijd vaak centraal. Dit optimale gebruik van middelen is ook belangrijk bij de controle van mechatronische systemen waar typisch ofwel de tijd ofwel het energieverbruik voor een taak geminimaliseerd wordt. Deze thesis ontwikkelt regelaars die de tijd minimaliseren om een punt-tot-punt beweging uit te voeren, dit wil zeggen een verkorting van de insteltijd van het systeem. In tegenstelling tot bestaande lineaire regelsystemen gebeurt de minimalisatie van de insteltijd in deze thesis voor willekeurige punt-tot-punt bewegingen en niet enkel voor één voorafbepaalde stap. Deze tijdsoptimale regelaars zijn ontworpen binnen het modelgebaseerde-regelaarraamwerk (MPC). Binnen het MPC raamwerk wordt het ingangssignaal bepaald door tijdens elke bemonsteringsperiode, zijnde de tijd tussen het aanleggen van twee ingangssignalen, een optimalisatieprobleem op te lossen. Het optimaliseren van het ingangssignaal binnen elke bemonsteringsperiode laat toe om expliciet de beperkingen van het systeem, zoals actuatorsaturatie, in rekening te brengen. De tijdsoptimale regelaars zijn ontworpen voor mechatronische systemen met bemonsteringsperioden van enkele milliseconden. Daarom is de snelle oplosbaarheid van de optimalisatieproblemen een belangrijke factor bij het ontwerp van deze regelaars. De belangrijkste bijdragen van dit doctoraat zijn als volgt. Als eerste is het minimaliseren van de insteltijd gedefinieerd als een optimalisatieprobleem binnen het MPC raamwerk. Vervolgens is de structuur van deze optimalisatieproblemen geanalyseerd en geëxploiteerd zodat deze problemen nu geformuleerd kunnen worden met een voldoende groot aantal variabelen om toepasbaar te zijn voor relevante mechatronische problemen en desalniettemin oplosbaar zijn binnen enkele milliseconden. Alle ontwikkelde regelaars zijn ook experimenteel gevalideerd op representatieve mechatronische testopstellingen zoals lineaire motoren en een portaalkraan. Deze experimentele validatie toont aan dat met de ontwikkelde regelaars bemonsteringsperiodes van enkele milliseconden haalbaar zijn en dat de insteltijd zeer sterk verminderd kan worden ten opzichte van lineaire regelaars en traditionele MPC regelaars.

Binnen dit globale kader van het minimaliseren van de insteltijd zijn drie regelaars ontworpen. Eerst is een voorwaarts gekoppelde regelaar ontworpen die een

referentietraject genereert dat de insteltijd minimaliseert voor punt-tot-punt bewegingen. Deze regelaar is ontworpen als een performanter alternatief voor bestaande voorwaarts gekoppelde regelaars. Vervolgens is een tijdsoptimale terugkoppelregelaar ontworpen. Het introduceren van terugkoppeling laat toe om storingen en statische fouten te compenseren. In een laatste stap is een regelschema voorgesteld dat tijdsoptimale regelaars combineert met lineaire regelaars. Dit regelschema laat toe om aan de vereisten van een industriële lineaire motor te voldoen, namelijk een kleine insteltijd gecombineerd met een instelnauwkeurigheid in het submicrometergebied.

Nomenclature

Within this thesis, the following nomenclature is used:

- real-time optimization: solution of the optimization problem within one sampling period
- embedded optimization: solution of the optimization problem on an embedded controller
- on-line optimization: solution of the optimization problem in real-time on an embedded controller
- off-line optimization: solution of the optimization problem before motion and hence without constraints on the required computational time
- feasible solution: solution of the optimization problem while respecting all inequality and equality constraints is possible

List of symbols

Abbreviations

EI	extra insensitive
FIR	finite impulse response
FRF	frequency response function
KKT	Karush-Kuhn-Tucker
LMI	linear matrix inequality
LTI	linear time invariant
MIMO	multiple-input multiple-output
MPC	model predictive control
PID	proportional integral derivative
QP	quadratic problem
SI	specified insensitivity
SISO	single-input single-output
TOMPC	time optimal model predictive control
ZV	zero vibration
ZVD	zero vibration derivative

Symbols

$\bullet_k$	value of $\bullet$ at time step k in the prediction horizon
$\bullet_l$	value of $\bullet$ at time step l in real time
$\bullet^\star$	optimized value of $\bullet$
$\bullet_{\mathbb{A}}$	values of $\bullet$ in the active set
$\bullet_{\mathbb{I}}$	value of $\bullet$ in the inactive set
Δu	differential system input
u	system input
x	system state
y	system output
s	dummy variables

u_{ref} : reference input
x_{ref} : reference state
y_{ref} : reference output
$f(x, u)$: system dynamics function
$g(x, u)$: system output function
$h(x, u)$: system constraints function
A : discrete time state matrix
B : discrete time input matrix
C : discrete time output matrix
D : discrete time feedthrough matrix
N : prediction horizon length in feasibility problem
N_{min} : minimal prediction horizon length in feasibility problem
N_{max} : maximal prediction horizon length in feasibility problem
K : prediction horizon length in mixed integer problem
n : number of system states
n_u : number of system inputs
n_y : number of system outputs
ζ : damping constant
λ : lagrange multiplier
C_o : controllability matrix
$\mathbb{X}$: feasible set

Contents

Dankwoord v

Abstract vii

Korte Inhoud ix

Nomenclature x

List of symbols xi

Contents xiii

List of Figures xvii

List of Tables xxiii

1 Introduction 1

1.1 Control of mechatronic systems 1

1.1.1 General control of mechatronic systems 1

1.1.2 Prefilters 2

1.1.3 Trajectory generation 6

1.1.4 General advantages and disadvantages of input shaping and trajectory generation 8

1.2 Model predictive control 9
1.2.1 General idea of model predictive control 9
1.2.2 Developments in MPC 11
1.2.3 MPC for fast applications 12
1.2.4 Time optimal MPC approaches 15
1.2.5 General review on model predictive control for mechatronic applications 16
1.3 Motivation 16
1.4 Chapter-by-chapter overview and contributions 18
1.4.1 Chapter-by-chapter overview 18
1.4.2 Main contributions 19

2 Predictive prefilter **21**
2.1 The predictive prefilter 21
2.1.1 Formulation of time-optimal problems in an MPC framework 22
2.1.2 Design of the predictive prefilter 23
2.1.3 Time optimality 24
2.2 Solution strategy: L_1 penalty 25
2.3 Implementation 27
2.4 Robustness 29
2.5 Validation 30
2.5.1 Test setup 30
2.5.2 Benchmark test 32
2.5.3 Validation and comparison with traditional prefilters by simulation 32
2.5.4 Validation and comparison with traditional prefilters by experiment 36
2.6 Conclusion 38

3 Time optimal MPC 41

3.1 Time optimal MPC controller design 41

3.1.1 TOMPC problem formulation 42

3.1.2 Proof of asymptotic convergence 44

3.1.3 Choice of $N_{\min}$ 45

3.2 Real-time Implementation 47

3.2.1 Problem formulation 47

3.2.2 Efficient reformulations 49

3.2.3 Efficient extension of the prediction horizon 53

3.3 Validation 53

3.3.1 Linear motor drive 54

3.3.2 Overhead Crane 61

3.4 Conclusions 71

4 Industrial application: combining TOMPC with high positioning accuracy 75

4.1 Problem statement 75

4.2 TOMPC with high positioning accuracy 78

4.2.1 New control scheme for the time optimal controller 78

4.2.2 Proof of asymptotic convergence and constraint satisfaction 79

4.2.3 Experimental validation 80

4.3 TOMPC with high positioning accuracy: scheme 2 84

4.4 Conclusion 86

5 Conclusions and future work 91

5.1 Conclusions 91

5.2 Recommendations for future work 92

A Implementation of the TOMPC controller 97

Bibliography **101**

Curriculum vitae **115**

List of publications **117**

List of Figures

1.1 General idea of inputshaping 4

1.2 Sensitivity curve for three different input shaping approaches showing the robustness with respect to uncertainty on the system eigenfrequency. This is shown for the ZV (black solid line), ZVD (grey solid line) and EI (black dashed line) approach. 6

1.3 General idea of MPC. At each time step l, the state $\bar{x}_l$ of the system is measured or estimated. Then, the optimal input is computed such that an objective function is optimized over the prediction horizon N. Finally, the first input u_0 is applied to the system. 9

2.1 General idea of a prefilter. The prefilter filters the reference r into an input u for the system resulting in output y. 22

2.2 Picture and schematic drawing of the mass-spring-damper test setup. 31

2.3 Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line), for a full step reference (black dashed line). Note that both lines coincide. 33

2.4 Input of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line) for a worst case scenario reference step. 33

2.5 Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). The reference is not one big step, but a series of smaller steps (black dashed line). 34

2.6 Input of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line) for a series of step references different from the worst case scenario reference step. 34

2.7 Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). The reference is a series of three steps (black dashed line), where a new setpoint is requested before the previous setpoint is attained. 35

2.8 Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). The reference is a series of two steps (black dashed line), where the second step is reversed compared with the first step. 35

2.9 Experimental validation of the predictive input shaping prefilter design: reference motion (black dashed line), and system response obtained by prefiltering with a classical input shaping prefilter (grey solid line) and with the predictive prefilter (black solid line). 37

2.10 Optimal experimental input to the system computed by the predictive prefilter. 37

2.11 Output of a perturbed system (grey solid line) and the nominal system (black solid line) for a reference step (black dashed line) filtered with a non-robust prefilter. 38

2.12 Robust design experiment: a reference step (black dashed line) is applied to a perturbed system with a 20 % change in eigenfrequency. This reference is robustly prefiltered and the resulting response (black solid line) is still free of vibrations. 39

2.13 Optimal experimental input to the system computed by the robust predictive prefilter. 39

3.1 Visualization of level sets of $V_B^\star$ in $\bar{x}_l$ space 44

3.2 The 1 % settling time expressed as number of sample times for the second order system presented in Chapter 4 as a function of the lower bound on the settling time N_{min}. 46

3.3 Required computational time in Matlab for four different methods to solve the TOMPC optimization problem 52

3.4 Picture of the linear motor drive test-setup in the PMA-lab at KULeuven. 54

3.5 Measured (black solid line) and identified (grey solid line) FRF of the linear drive system. 55

3.6 Output comparison on a reference step of 10 cm between TOMPC and a linear controller which is optimized for a step of 10 cm. . . . 57

3.7 Input comparison on a reference step of 10 cm between TOMPC and a linear controller which is optimized for a step of 10 cm. 57

3.8 Output comparison on a reference step of 5 cm between TOMPC and a linear controller which is optimized for a step of 10 cm. 58

3.9 Input comparison on a reference step of 10 cm between TOMPC and a linear controller which is optimized for a step of 5 cm. 58

3.10 Linear drive system: Drive system motion obtained with TOMPC (black solid line) and MPC (grey solid line) for a reference step of 10 cm (black dashed line) 59

3.11 Linear drive system: Relative output error obtained by control with TOMPC (black line) and MPC (grey line) for a reference step of 10 cm 60

3.12 Linear drive system: Control input signal obtained by control with TOMPC (black line) and MPC (grey line) for a reference step of 10 cm 60

3.13 Linear drive system: Control input applied to the system by using a TOMPC controller with $N_{\min} = 8$ (black line) and $N_{\min} = 6$ (grey line). 61

3.14 Linear drive system: Drive system motion (full line) obtained when a reference trajectory of two steps is requested (dashed line). The second step is requested before the first one is completed. 62

3.15 Linear drive system: Control input obtained when a reference trajectory of two steps is requested and the second step is requested before the first one is completed. 62

3.16 The overhead crane in the PMA-lab at KULeuven. 63

3.17 Schematic representation of the overhead crane 64

3.18 Measured FRF (black line) and FRF of the identified model (grey line) relating the input u to the position of the trolley x 65

3.19 Measured FRF (black line) and FRF of the identified model (grey line) relating the input u to the swing angle θ. 66

3.20 Position of the trolley (black line) controlled by a TOMPC controller for a reference step of 10 cm (grey line). 67

3.21 Swing angle of the TOMPC controlled system for a reference step of 10 cm. 67

3.22 Input applied to the overhead crane for a requested step of 10 cm controlled by a TOMPC controller. 68

3.23 Differential input applied to the overhead crane for a requested step of 10 cm controlled by a TOMPC controller. 68

3.24 Position of the trolley (black line) controlled around the zero position (grey line) by a TOMPC controller when a disturbance is applied. . 69

3.25 Swing angle of the TOMPC controlled system when a disturbance is applied. In the grey zone, the system is manually moved out of its reference position. 69

3.26 Input applied to the overhead crane controlled by a TOMPC controller when a disturbance is applied. 70

3.27 Differential input applied to the overhead crane controlled by a TOMPC controller when a disturbance is applied. 70

3.28 Position of the trolley (black line) controlled by a TOMPC controller for a series of reference steps (grey line). 71

3.29 Swing angle of the TOMPC controlled system for a series of reference steps. 71

3.30 Input applied to the overhead crane controlled by a TOMPC controller for a series of reference steps. 72

3.31 Differential input applied to the overhead crane controlled by a TOMPC controller for a series of reference steps. 72

4.1 Experimental test setup of a linear motor with airbearings. 76

4.2 Measured FRF (black) and FRF of the identified model (grey) from the current input [A] to the position of the slider [m]. 77

4.3 Graphical representation of the benchmark requirements on a 25 mm reference step for the linear motor test setup at ETEL. 78

4.4 Switching control scheme which combines time-optimal control with high-accuracy control. 79

4.5 Implementation of the time-optimal high-accuracy control scheme in Simulink. 81

4.6 Position of the linear motor drive (black solid line) and the reference trajectory (grey solid line) for a reference step of 25 mm at time 0.1 s. 82

4.7 Logarithmic output error of the position of the linear motor with the time-optimal high-accuracy control scheme. 82

4.8 Logarithmic output error of the position of the linear motor with the time-optimal high-accuracy control scheme. Output error given for three different controllers in order to illustrate robustness. . . . 83

4.9 Control input current of the linear motor actuators when a step reference of 25 mm is requested at time 0.1 s and the switching between TOMPC and PID control occurs at time 0.24 s. 84

4.10 Adapted control scheme developed at KULeuven. The time optimal controller runs at 250 Hz. The PID controller and the simulator $\tilde{G}$ run both at 12 kHz. 85

4.11 Total control input $(u_{\text{ff}} + u_{\text{fb}})$ (grey solid line) and feedforward input (u_{ff}) (black solid line) for a reference step of 25 mm imposed at time 0.1 s, obtained using the control scheme of Fig. 4.10 when the system simulator $\tilde{G}$ runs at 250 Hz. 85

4.12 Output comparison of the two developed time-optimal high-accuracy control schemes. 87

4.13 Logarithmic output error comparison of the two developed time-optimal high-accuracy control schemes. 87

4.14 Input of the feedforward time-optimal high-accuracy control schemes. 88

4.15 Input of the time-optimal high-accuracy switching control schemes presented in Section 4.2. 88

List of Tables

2.1 Poles of the fifth order system 32

2.2 Required on-line computation time for the robust and non-robust predictive prefilter. A maximal computation time of 10 ms is allowable because of sampling period constraints. 40

3.1 Relative comparison of worst case computation times in Matlab for the linear motor drive system discussed in Section 3.3 for a reference step of 10 cm, using four different solution methods. 52

3.2 The 1% settling time for different steps using TOMPC and MPC . . 59

Chapter 1

Introduction

This doctoral thesis discusses the design, implementation and experimental validation of time-optimal controllers for point-to-point motion of LTI mechatronic systems. These controllers are developed within the model predictive control (MPC) framework. This chapter motivates the research discussed in this thesis. First, Section 1.1 situates the controller design within the general context of control of mechatronic systems and gives an overview of the current state-of-the-art of linear approaches to obtain near time-optimal motion for the considered class of mechatronic systems. Second, Section 1.2 presents the concept of MPC and discusses current research on the application of MPC to mechatronic systems. Third, Section 1.3 states the problem formulation and hypotheses of this doctoral thesis. The chapter ends with a chapter-by-chapter overview and a summary of the main contributions in Section 1.4.

1.1 Control of mechatronic systems

1.1.1 General control of mechatronic systems

Mechatronic systems [Buur, 1990, Tomizuka, 2002] are systems which integrate actuators, sensors, controls and mechanical design. By integrating all these aspects during the system design, much higher control performance can be obtained than if all these aspects are considered as add-ons to each other. This definition of mechatronic systems encompasses a broad range of systems like robots, linear motors, planes, cars, ... By externalizing nonlinear dynamics as e.g. friction, many of these systems can be represented by linear time invariant (LTI) models, i.e. by system models with constant linear system dynamics. For these systems, typically three kinds of control problems can be defined. First, there are tracking control problems which follow a given trajectory in state space and time coordinates under a given criterion as accurately as possible. Second, there are control problems where a trajectory in state space coordinates has to be followed and the corresponding time frame is not specified. These controllers can be optimized with respect to e.g. the required path following time, energy consumption or accuracy. Third, there are regulation control problems which have as an objective to arrive at or stay around a given reference position without specifying constraints on the underlying trajectories. This regulating controller can be optimized with respect to e.g. settling time, overshoot and energy consumption. This thesis focuses on the regulation control problem for LTI systems with the objective to minimize the settling time, i.e. the system has to settle as fast as possible at the desired setpoint while respecting the system constraints. In order to obtain true time-optimal control, system constraints must be active, i.e. the system will work at its input or output limits.

Most LTI mechatronic systems are currently still controlled using linear feedback controllers such as traditional PID controllers tuned by e.g heuristic Ziegler-Nichols rules [Ziegler and Nichols, 1942, Franklin et al., 2001] or more advanced model-based controllers, like e.g. H_∞ robust controllers [Zhou et al., 1995] or internal model controllers [Morari and Zafiriou, 1989]. The main disadvantage of these linear controllers is that they can not account easily for constraints on inputs, outputs and states or only by introducing a lot of conservativeness [Scherer et al., 1997]. For numerous applications, these controllers are perfectly suited and they can be well-tuned. However, for regulation problems where minimization of the settling time is requested within stringent input constraints, they do not perform well. In order to obtain near time-optimal control, numerous linear approaches exist; namely prefilters which filter the reference step in order to cancel the vibrations induced on the eigenfrequencies of the system, and trajectory generators which replace the reference step by a smoother reference trajectory which induces less residual vibrations. Therefore the following sections give an overview of prefilters (Section 1.1.2) and trajectory generators (Section 1.1.3).

1.1.2 Prefilters

Basic input shaping

Input shaping [Smith, 1957, Singer, 1989, Singer and Seering, 1990, Singhose et al., 1994] is a prefilter technique which transforms a reference step into a reference trajectory such that the system reaches the desired setpoint without residual vibrations, i.e. once the system arrives at the setpoint there are no vibrations around this setpoint, thereby reducing the settling time. The original input shaping prefilter is a continuous time filter which convolves the requested step input with a finite-impulse-response (FIR) filter, see Fig. 1.1, where the amplitude f_k and the position t_k of the impulses are determined analytically based on the vibration equation of a second order system. The impulses are chosen such that they compensate the dominant second order poles of the system, i.e. they are a solution of the following set of equations:

$$\sum_{k=0}^{K} f_k \mathrm{e}^{-\zeta\omega_n(t_K - t_k)} \sin(t_k \omega_n \sqrt{1-\zeta^2}) = 0, \tag{1.1a}$$

$$\sum_{k=0}^{K} f_k \mathrm{e}^{-\zeta\omega_n(t_K - t_k)} \cos(t_k \omega_n \sqrt{1-\zeta^2}) = 0, \tag{1.1b}$$

where ω_n is the eigenfrequency and ζ is the damping constant of the system. It can be shown that these prefilters are comparable to notch filters [Murphy and Watanabe, 1992]. In order to obtain the shortest possible settling time, the length of the prefilter is minimized, i.e. the position t_K of the last impulse of the FIR filter is placed as early as possible. In the original input shaping prefilter design only positive impulses f_k are allowed, thereby guaranteeing input constraint satisfaction if the unfiltered reference step satisfies the input constraints [Singer and Seering, 1990]. However, this basic approach is sensitive to model-plant mismatch and has the disadvantage of a rather long filter length.

Extensions to decrease settling time

In order to decrease the settling time of input shapers, most extensions to the original input shaping prefilter allow not only positive but also negative impulses f_k [Singhose et al., 1997]. This reduces the filter length significantly, i.e. the position of the last impulse t_K, and hence the settling time. However in contrast to the original input shaping design with only positive impulses, input constraint satisfaction is not longer guaranteed for all step references which do not saturate without prefiltering. Also, the robustness of the input shaping prefilter decreases further in comparison with the original input shaping prefilter with only positive impulses. Therefore, the effect of saturation and other non-linearities is analyzed and it is shown how these effects can be mitigated [Sorensen and Singhose, 2007, Sorensen et al., 2008] by limiting the drivable region or by preconditioning the reference signals. This introduces conservativeness as also non-saturating references are preconditioned. Other extensions improve the total settling time by incorporating knowledge about the reference step [Baumgart and Pao, 2007, Robertson and Erwin, 2007] which is by definition only optimal for one specific trajectory however. Kenison and Singhose [2002] improve the total system performance as they design the closed-loop controller taking into account that an input shaping filter which filters the step input is added to the closed-loop system .

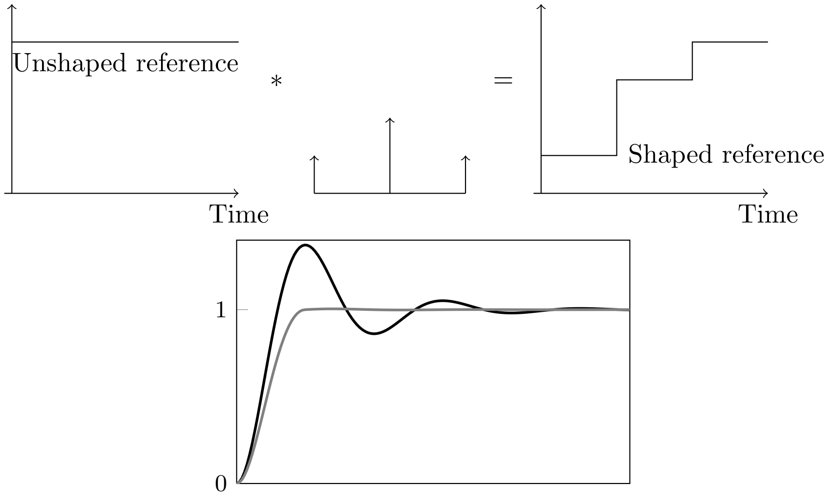

Figure 1.1: General idea of inputshaping. An unshaped reference step is convolved with a FIR filter in order to obtain a shaped reference which is applied to the system. This results in an output behavior (grey line) with typically a higher rise time but much lower settling time than the unshaped response (black line).

Extensions for higher order systems

Input shaping is originally developed for second-order systems. Higher order modes can however also contribute significantly to residual vibrations. Therefore, research has been conducted on the extension of input shaping to higher order systems [Singhose et al., 1997, La-orpacharapan and Pao, 2004, 2005]. One approach convolves multiple low order input shaping filters which each solve problem (1.1) for one vibration mode. However, this approach introduces large filter delays as the length of the total prefilter equals the sum of the length of the original filters minus the number of filters. Other approaches based on the original input shaping prefilters, solve the extended analytical problem (1.1) for all modes simultaneously, thereby reducing the length of the total prefilter and hence the settling time. However, the resulting optimization problems grow increasingly complex and are non-convex [Boyd and Vandeberghe, 2004]. Sungyung et al. [1999], Van den Broeck et al. [2008] reformulate the input shaping problem for discrete time systems based on the dynamic equations of the system: $x_{k+1} = Ax_k + Bu_k$. This reformulation casts the determination of the optimal impulses of the FIR filter into a linear optimization problem which guarantees a fast solution. The problem formulation which characterizes this technique is independent of the order of the system. Moreover, constraints on the inputs and outputs of the system can be included easily within the linear optimization framework. Singh [2010] also presents a numeric approach which allows to minimize the settling time.

Extensions for robustness

The basic input shaping prefilters are very sensitive with respect to the value of the eigenfrequency of the system, i.e. if the real eigenfrequency of the system differs even only moderately from the nominal eigenfrequency used in the prefilter design, the system still exhibits large residual vibrations if the system damping ζ is small, despite the prefiltering of the reference signal. Robustness with respect to parameter uncertainty is for input shapers typically expressed by sensitivity curves. These curves show the residual vibration of a system on a filtered reference step in function of a normalized parameter, i.e. the actual value of the parameter divided by the nominal identified value of the parameter. For input shapers, this sensitivity is typically expressed as a function of the system eigenfrequency. In the original zero vibration (ZV) prefilter design, the sensitivity curve is zero for the nominal eigenfrequency. However, this sensitivity curve increases fast if the eigenfrequency differs from the nominal value, see Fig. 1.2. To introduce robustness, this sensitivity curve should have low values over a wide range of eigenfrequencies. This robustness can be introduced by imposing extra constraints locally or globally on the sensitivity curve. Locally, the prefilter design can not only constrain the value of the sensitivity curve for the nominal system but also the derivatives of this curve, thereby effectively pushing down the sensitivity curve locally. This technique is implemented in e.g. the zero vibration derivative (ZVD) prefilter [Singer and Seering, 1990]. Other basic robust approaches called extra insensitive (EI) filters, obtain robustness by imposing global constraints on the sensitivity curve. These EI prefilters require a limited residual vibration for multiple points on the sensitivity curve but do not impose zero vibration for the nominal system as they assume that the real value of the system parameters is never exactly known [Singhose et al., 1994]. Fig. 1.2 shows the sensitivity curve for the three basic input shaping approaches; i.e. ZV (black solid line), ZVD (grey solid line) and EI (black dashed line). The EI approach tends to have a broader insensitivity range than the ZVD approach [Vaughan et al., 2008a,b]. Van den Broeck et al. [2008] introduce robustness in their linear programming framework, analogous to the EI approach, by minimizing the residual vibrations on a step reference not only for the nominal system but for multiple systems with different system parameter values. Another robust approach, called the specified insensitivity (SI) method, allows to impose a desired insensitivity [Singhose et al., 1996]. This SI filter generates the highest robustness for a given filter length. Moreover, this design approach can also generate asymmetric sensitivity curves. A disadvantage of SI filters is that no closed form expression exists. Further extensions of robust prefilters include knowledge on the distribution of the uncertainty, thereby minimizing the expected residual vibrations [Pao and Lau, 2000]. A disadvantage of all robust input shapers is their increase in filter length and correspondingly the settling

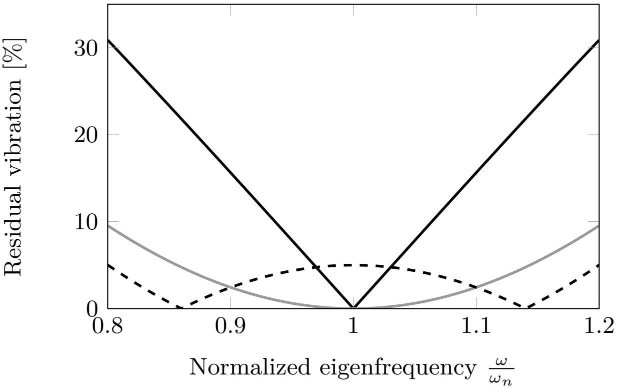

Figure 1.2: Sensitivity curve for three different input shaping approaches showing the robustness with respect to uncertainty on the system eigenfrequency. This is shown for the ZV (black solid line), ZVD (grey solid line) and EI (black dashed line) approach.

time of the controlled system. A different approach therefore uses short non-robust filters which are adapted to the filter frequency which is identified during motion. These adaptive prefilters are developed both for repetitive [Park and Chang, 2001] and non-repetitive trajectories [Cutforth and Pao, 2004]. Their disadvantage is however their higher complexity, which requires more expensive hardware.

1.1.3 Trajectory generation

Off-line trajectory generation

As the application of pure reference steps to a system induces a lot of residual vibrations as illustrated in Fig. 1.1, these steps are often replaced by smoother reference trajectories which induce less or no residual vibrations because of their smoothness. It should be noted that the goal of the control design is to regulate the system at a given reference position. Hence, the underlying trajectory is not important, which allows design freedom on these trajectories. Therefore, the trajectory generators replace the reference step by a reference trajectory which has the same end point value but which is smoothened by imposing constraints on its derivatives. One very basic reference trajectory approach uses trapezoidal velocity profiles to smoothen the step reference, i.e. the reference trajectory accelerates linearly until the maximum velocity is reached and at the end of motion decelerates linearly until the velocity reaches zero. As these profiles yield infinite jerk, they still induce high residual vibrations [Kyriakopoulos and Saridis, 1994, Lin et al., 2002]. Therefore, slightly more advanced approaches called S-curves [Lewin, 1994] apply higher order polynomial trajectories; e.g. a ramp reference profile is imposed on the acceleration instead of on the velocity. The resulting residual vibrations can be minimized by selecting an optimal ramp-up time for the acceleration based on the assumption that the system can be represented by a two-mass-spring system [Meckl et al., 1998]. Instead of using symmetrical trajectories, also asymmetrical trajectories with faster acceleration and slower deceleration can be used to decrease settling time [Tsay and Lin, 2005]. Beazel and Meckl [2005], Chatlatanagulchai, Beazel, and Meckl [2006] use reference trajectories based on sine base functions which are tuned to the eigenfrequency of the system, instead of polynomial functions, thereby generating more robust reference trajectories for nonlinear systems with changing resonances. The former techniques limit only velocity, acceleration, and possibly higher derivatives of the reference trajectory. They can not take into account input constraints directly. Moreover, minimization of the settling time is only approximated by using smoother trajectories, which because of this smoothness introduce less vibrations and overshoot. Note that more advanced off-line design approaches allow to impose input constraints and can generate true time-optimal trajectories. Henrion and Lasserre [2006] design a reference trajectory which is defined as one polynomial. The coefficients of this polynomial are optimized by solving a linear matrix inequality (LMI). By increasing the degree of the polynomial, constraints on inputs and outputs can be imposed. Other approaches based on the combination of multiple cubic splines [Kwakernaak and Smit, 1968], guarantee smoothness by design. Demeulenaere, Pipeleers, De Caigny, Swevers, De Schutter, and Vandenberghe [2009b], Demeulenaere, De Caigny, Pipeleers, De Schutter, and Swevers [2009a] extend the approach of Kwakernaak and Smit [1968] to allow higher orders of smoothness. By solving a series of linear problems, these approaches find the time-optimal trajectory while taking system constraints into account.

On-line trajectory generation

With an increase in affordable computation power, the basic S-curve reference trajectories are now often computed on-line [Nguyen et al., 2008]. These approaches generally take constraints on velocity and acceleration into account [Zheng et al., 2009]. Rew, Ha, and Kim [2009] generate on-line asymmetric S-curves to further decrease the settling time. More advanced controllers also put constraints on higher derivatives of the motion trajectory and can take into account part of the system dynamics [Macfarlane and Croft, 2003, Lambrechts et al., 2005]. For pure double integrators, also approximate time-optimal controllers can be obtained (PTO) [Workman et al., 1987a,b, Workman and Franklin, 1988, Zanasi et al., 2000]. This approach has been extended to third order systems by Pao and Franklin [1993] and Zanasi and Morselli [2003]. A disadvantage of all these approaches is that they can not take input constraints into account directly. Moreover, minimization of settling time is not directly imposed; these approaches are usually based on smoothness of the reference trajectory and can at most guarantee time-optimality for simple low order systems such as a series of pure integrators.

1.1.4 General advantages and disadvantages of input shaping and trajectory generation

Trajectory generators and input shapers are powerful tools to reshape reference steps into reference trajectories which in comparison with a pure step reference, induce less residual vibrations at the end of motion. A big advantage of these techniques is their low on-line computational complexity, allowing an implementation on cheap hardware. They are successfully applied to applications in high precision positioning stages [Li et al., 2009, Amthor et al., 2010], machining applications [Olabi et al., 2010], wafer stages [DeRoover and Sperling, 1997], satellites [Tuttle and Seering, 1997], milling machines [Fortgang et al., 2005, Marquez et al., 2006], DC-DC converters [Yousefzadeh et al., 2008], coordinate measuring machines [Jones and Ulsoy, 1999] and cranes [Sorensen et al., 2007, Huey et al., 2008, Vaughan and Singhose, 2009, Vaughan et al., 2010]. These methods can be divided into two groups depending on whether the reference step is transformed during motion or before motion. The approaches which transform the reference step during motion are usefull for applications where steps over a wide range have to be processed and time-optimality is less an issue. However, they are conservative with respect to settling time for reference steps which differ from the worst-case, where the worst-case is usually defined as the largest possible displacement. Moreover, often combinations of reference steps exist which will violate the system constraints. Trajectory generators which generate reference trajectories before motion, can guarantee constraint satisfaction and time-optimality for all reference steps. However, as this reference trajectory has to be computed beforehand, they are only efficient if a limited set of known references is imposed. If these systems have to react on new reference steps, a new reference trajectory has to be computed. This introduces a delay between the step request and the start of system control action. This delay can easily be as large as the time required for performing the reference step.

Hence, for applications which require time-optimal control over a wide range of unknown reference steps, these approaches are not suitable. For these applications control approaches which compute on-line an optimal system reference and which can account for actuator constraints, are required. This however comes with the cost of more expensive hardware.

1.2 Model predictive control

The controllers in this thesis are developed within the MPC framework. Therefore, Section 1.2.1 and Section 1.2.2 give a short general introduction to MPC. Section 1.2.3 and Section 1.2.4 discuss current applications of MPC in the field of mechatronics together with time-optimal control techniques.

1.2.1 General idea of model predictive control

MPC is an advanced control technique that originated in the sixties and gained popularity in the eighties [Richalet et al., 1978, García et al., 1989]. The MPC control approach is described in the excellent textbook [Maciejowski, 2000] and paper [Rawlings, 2000]. An MPC controller computes the optimal system input on-line by solving an optimization problem and taking into account the system constraints. The MPC controller first measures or estimates the current state of the system $\bar{x}_l$. Then, the MPC controller determines the optimal open-loop system input u_k over a given horizon N, taking into account the system limitations. When the optimal input is determined, the first part of this optimal input u_0 is applied to the system. The next sampling time, the whole procedure is repeated, thereby introducing system feedback. Figure 1.3 illustrates the MPC-idea.

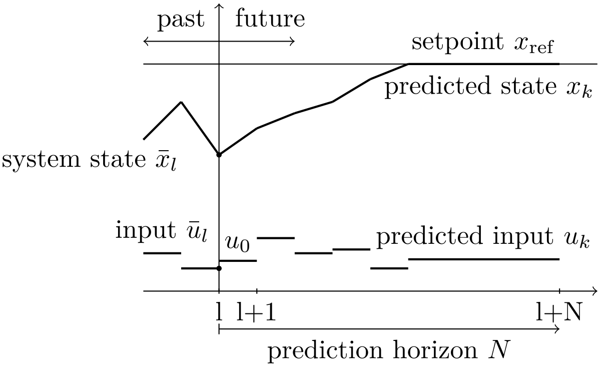

Figure 1.3: General idea of MPC. At each time step l, the state $\bar{x}_l$ of the system is measured or estimated. Then, the optimal input is computed such that an objective function is optimized over the prediction horizon N. Finally, the first input u_0 is applied to the system.

Traditionally, the following optimization problem is to be solved at every sampling time:

$$V_A^\star(\bar{x}_l, N) = \min_{x,u} \sum_{k=0}^{N-1} \|u_k - u_\text{ref}\|_R^2 + \|x_k - x_\text{ref}\|_Q^2 + E(x_N) \tag{1.2a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_l, \tag{1.2b}$$

$$x_{k+1} = f(x_k, u_k), \tag{1.2c}$$

$$0 \leqslant g(x_k, u_k), \quad k \in [0, N-1], \tag{1.2d}$$

$$x_N \in \mathbb{T}, \tag{1.2e}$$

where x_k, u_k are the optimization variables over the prediction horizon N. They respectively represent the system states x and the system inputs u at time step k in the prediction horizon. $\bar{x}_l$ is the measured or estimated system state at time l. The optimization uses a system model to predict the system behavior for a sequence of control variables (1.2b)–(1.2c) and accounts for bounds on inputs, outputs and internal states by inequality constraints (1.2d). The input is optimized considering the Euclidean norm of the deviation of the system input u_k and system state x_k from their references u_ref and x_ref (1.2a). These deviations are weighted with the positive definite matrices R and Q respectively. Often an extra terminal state weight (1.2a) or terminal state constraint (1.2e) is added to compensate for the finiteness of the horizon N. As the MPC controller has to solve optimization problem (1.2) every sampling time, MPC first became popular in the process and chemical industry [Qin and Badgwell, 1996, 2003], where sampling periods in the order of seconds or minutes allow more computation time. The required computation time strongly depends on the problem size and hence on the length of the horizon N, the number of system states, inputs and outputs and the number of constraints. Since the eighties various topics on MPC have been analyzed, both theoretically and practically, see e.g. [Morari and Lee, 1999, Mayne et al., 2000] and the textbook [Rawlings and Mayne, 2009]. The first analysis on MPC was mostly theoretical and focused on stability and robustness issues. Therefore, these topics are covered in Section 1.2.2. Gradually, research focus shifted to also take into account faster solvability of the optimization problems. Research dedicated to the solution of MPC problems in real-time for fast applications and the application of MPC to mechatronic systems are discussed in 1.2.3.

1.2.2 Developments in MPC

Stability

The concept of MPC originated from industrial applications under various names as e.g. dynamic matrix control [Cutler and Ramaker, 1980] and generalized predictive control [Clarke et al., 1987, Bitmead et al., 1990]. Therefore, early academic research examined how and why MPC controllers perform well and how stability can be guaranteed, see e.g. [Keerthi and Gilbert, 1988, Rawlings and Muske, 1993, Scokaert and Rawlings, 1998, Chen and Allgöwer, 1998a,b, Mayne et al., 2000, Limon et al., 2003]. These approaches to introduce stability are mostly based on introducing a terminal constraint and/or a weight on the terminal state. The terminal constraint forces the system to enter a set where a local controller is able to stabilize the system. The weight on the terminal state introduces an approximation of the infinite horizon cost or more specifically a control Lyapunov function [Mayne et al., 2000].

Robustness

An MPC controller which is designed for the nominal identified system often already has inherent robustness properties [Marruedo et al., 2002]. Since this inherent robustness is not always sufficient, research also focuses on the robustness of MPC controllers. Various approaches based on min-max problems already exist for systems with limited disturbances [Lee and Yu, 1997, Scokaert and Mayne, 1998]. To decrease the prohibitive computational cost of solving the corresponding min-max optimization problem, these min-max problems are approximated based on control strategies using feedback policies [Langson et al., 2004, Ramirez et al., 2006]. E.g. Kothare, Balakrishnan, and Morari [1996] reformulate the min-max problem as an LMI optimization problem. Alamo, Ramirez, de la Peña, and Camacho [2007] introduce an approximation resulting in a quadratic program (QP) solvable in real-time for industrial applications. Goulart et al. [2006, 2008] show that a robust MPC controller based on full state feedback, can be obtained by the solution at every sampling time of only one convex QP of which the structure can be exploited to speed up the computation.

Offset-free control

Muske and Badgwell [2002], Pannocchia and Rawlings [2003] and Pannocchia [2004] on the other hand discuss how exact setpoint tracking is obtained in the presence of disturbances. They introduce a disturbance model and show under which conditions of observability and controllability these approaches deliver off-set free tracking. One of their conditions is that the number of additional disturbance states has to be equal to the number of measured outputs. Hence, for systems with many measured outputs, the dimensions of the internal model and correspondingly the computational load increase significantly. Maeder, Borrelli, and Morari [2009] show how the state estimator can be designed such that the number of disturbance states can be reduced to the number of desired tracking outputs. It should be noted that offset-free control often also can be obtained by defining optimization problem (1.2) in terms of variation of the input Δu instead of the input u [Morari and Lee, 1999].

1.2.3 MPC for fast applications

Thanks to the increase in computation power and the development of more advanced algorithms, MPC is nowadays starting to become applicable to faster applications such as e.g. motor engines with sampling times in the millisecond range. However, their application is not yet widespread. To solve the optimization problems in these small sampling periods, the problem structure needs to be exploited maximally.

Explicit Model Predictive Control

A very successfull approach to solve MPC problems in real-time for fast applications is explicit model predictive control [Pistikopoulos et al., 2000, Bemporad et al., 2002a,b]. Explicit MPC controllers are originally developed for linear systems. They exploit the property that the solution of the MPC optimization problem is a continuous piecewise affine function of the state on polytopic, so called critical, regions [Klatte, 1979, Zafiriou, 1990], i.e. locally the optimal control law is given by: $u^\star = Kx + b$. In the controller design, all computational effort is performed beforehand and the optimal controller for each critical region is stored. On-line, the current state is estimated and the corresponding optimal control law looked up in the memory. The basic algorithm is adapted by Borrelli et al. [2003] to obtain a more efficient off-line computation of the local controllers. Tondel et al. [2003] developed an explicit MPC controller which searches the optimal solution in real-time more efficiently. This approach has been applied successfully to e.g. PWM converters [Mariethoz and Morari, 2009], diesel engines [Ortner and del Re, 2007] and automatic cruise control [Naus et al., 2010]. Also, Johansen [2004] uses an explicit approximating controller for the embedded control of nonlinear systems, a technique which is further extended to incorporate robustness [Grancharova and Johansen, 2009]. The main disadvantage of these controllers, however, is their limitation to applications with a low-dimensional state space and horizons because of the exponential increase in complexity and storage requirements. Jones and Morari [2010] alleviate this problem partly by making a trade-off between suboptimality and complexity. Zeilinger, Jones, and Morari [2011] on the other hand use the explicit controllers as an hotstart method for on-line MPC methods as presented in the next subsection. Canale, Fagiano, and Milanese [2009] reduce the on-line load by computing on-line only an approximated solution instead of the optimal solution.

Efficient numerical algorithms for fast MPC

To overcome the limitations of explicit MPC, fast MPC approaches solve the underlying optimization problem (1.2) on-line, and they exploit the system structure and problem similarities in the sequence of MPC problems maximally. An advantage of these methods is that they allow larger horizons and higher dimensionsal systems. However, a disadvantage of these systems is that less or no guarantees can be given on the required computation time. A first set of methods solve optimization problem (1.2) using active set methods. These methods are typically more performant for high dimensional systems with relatively short prediction horizons. Ferreau, Bock, and Diehl [2008] solve a condensed version of (1.2). Using the solution of the previous time sample, they hotstart the solution procedure. This increases the solution speed which allows sampling times in the range of tens of milliseconds [Ferreau et al., 2007]. Wills, Bates, Fleming, Ninness, and Moheimani [2005] also solve the MPC optimization problem on-line using an active set solver. They obtain sampling frequencies up to 25 kilohertz. A second set of methods solve optimization problem (1.2) by using an interior point solver. These methods are typically more performant for low dimensional systems with long prediction horizons. E.g. Wang and Boyd [2010] solve the uncondensed version of optimization problem (1.2) by using an interior point solver. Shahzad, Kerrigan, and Constantinides [2010] present a method which also solves the MPC problem with an interior point solver, however by better conditioning the problem, the rate of convergence and hence solution speed is increased.

Other approaches to further increase the on-line computation speed, exploit the fact that at every time sample, an optimization problem with the same structure has to be solved. Mattingley and Boyd [2009] present a code generator which generates off-line an optimized optimization procedure for convex problems. The hereby generated code maximally exploits the underlying problem structure for solution of QP's by an interior point solver. However, the currently attainable problem dimensions are still limited. Houska, Ferreau, and Diehl [2010] extend this idea by generating code not only for the QP-solver but also for system integration and condensing, thereby generating optimized code for solving nonlinear MPC optimization problems. Earlier, Seguchi and Ohtsuka [2003], Ohtsuka [2004] developed an automatic code generator AutoGenU which solves nonlinear optimization problems using the GMRES algorithm. Moreover, nowadays research is also dedicated to exploit parallelism and the hardware structure optimally to increase the computational speed [Jerez et al., 2010, Wills et al., 2010].

Application of MPC to mechatronic systems

MPC is extensively used for applications with slow system dynamics in e.g. the process industry [Qin and Badgwell, 2003]. Nowadays MPC is also applied to mechatronic systems. However, the number of real mechatronic MPC applications is still very limited. Moreover, these applications are characterized by either sampling times in at least tens of milliseconds or by problems with limited dimensions. Alamir and Murilo [2008] use an MPC controller to control a double inverted pendulum. This controller considers only three possible inputs, thereby limiting the on-line load considerably. Also, the control structure is exploited for one specific application and can not be transferred to other applications. Miller, Kolmanovsky, Gilbert, and Washabaugh [2000], Casavola, Mosca, and Papini [2004] and Susanu and Dumur [2006] use a low load version of true predictive controllers, the reference governor [Gilbert et al., 1995, Bemporad et al., 1997, Bemporad, 1998], for applications as an electrostatically shaped membrane, inverted pendulum and milling machine. These controllers reshape a reference trajectory for a closed-loop system such that the internal controller does not saturate its actuators. The advantage of these techniques is that stability can easily be proven as they are based on traditional proofs of linear systems. However, they introduce conservativeness as the input is not controlled directly. Ferreau, Ortner, Langthaler, del Re, and Diehl [2007], Ortner and del Re [2007] implement an MPC controller on a diesel engine with a sampling time of fifty milliseconds, with an on-line and explicit approach. For this application they consider non-linear models. However, the considered optimization problems have a very short horizon in order to keep the problem solvable in real-time. Hence, this introduces conservativeness. Wills, Bates, Fleming, Ninness, and Moheimani [2005, 2008] implement an MPC controller on a flexible beam for vibrationcontrol. Their controller runs at a sampling frequency of five kilohertz but has only twelve decision variables. Coen, Saeys, Missotten, and Baerdemaeker [2008] use an MPC controller based on an active set method for the cruise control of a combined harvester, taking into account constraints on the input, rounds per minute and velocity of the harvester. This controller runs at a sampling frequency of twenty hertz and has ten decision variables. Naus, Ploeg, Van de Molengraft, Heemels, and Steinbuch [2010] implement an adaptive cruise control for trucks and cars where they also take into account safety and drivers comfort. Their MPC controller is based on the explicit MPC approach. Also Corona and De Schutter [2008] implement an adaptive cruise control through MPC using both an on-line Cplex solver and an explicit approach. This controller runs at a sampling frequency of one hertz.

For all these mechatronic systems, MPC is applied successfully. However, the considered applications allow only a very limited number of decision variables and sampling times above ten milliseconds. Moreover, minimization of settling time is never explicitly targeted in these control problems. The approach developed in this thesis allows to decrease the length of the sampling periods down to four milliseconds and allows to increase the number of decision variables up to forty-five.

1.2.4 Time optimal MPC approaches

In the field of optimal control [Pontryagin et al., 1962, Bryson and Ho, 1969], time-optimality has been an active topic of research. Zadeh [1962] presents how time-optimal problems can be formulated as a feasibility search based on linear problems (LP). Bashein [1971] decreases the required computation time by using an improved simplex algorithm, a technique which is employed by e.g. Scott [1986]. Kim and Engell [1994] propose a method to increase the computational speed further by starting the feasibility search with an optimal guess. These methods have been used to generate feedforward signals for e.g. unmanned aerial vehicles [Kim and Tilbury, 2001], flexible joints [Consolini and Piazzi, 2009] and high-rise elevators [Schlemmer and Agrawal, 2002]. The presented optimization approaches are too slow for real-time application however, and hence use of these approaches in a feedback configuration is not possible.

Zhao, Diehl, Longman, Bock, and Schlöder [2004] on the other hand develop a controller for robots with time-optimality as their objective; they work in continuous time and use a rescaling of time with discretized controls for their numerical solution instead of a feasibility search, making convergence proofs difficult. Similarly, Kirches, Sager, Bock, and Schloeder [2010] present a time-optimal control problem for automobiles with gearshifting. For these applications, solution times in the range of seconds are attainable. The properties of time-optimal control laws have also been analyzed theoretically [Kostyukova and Kostina, 2003, Maurer et al., 2005, Kostyukova and Kostina, 2009]. True time-optimal approaches related to MPC also exist for robot path-tracking problems. These methods minimize the time required to track a given trajectory in spatial coordinates [Verscheure et al., 2009a,b].

Other methods obtain time-optimal control by dividing the state-space into critical regions. Desoer and Wing [1961] developed one of the first approaches which was due to the representation of these critical regions difficult to implement however. Workman et al. [1987b] present a time-optimal feedback controller for double integrators. This method divides the state-space into two regions where the system actuators provide either maximal or minimal input. In order to account for measurement noise and small disturbances, they introduce a zone around the setpoint where a local, less agressive controller takes over the control action. This controller has been used succesfully for e.g. hard disk drives [Zhou et al., 2001]. Keerthi and Gilbert [1987] extend this method to higher order systems. Moreover, they allow not only control constraints but also state constraints. In order to account for disturbances, Mayne and Schroeder [1997] extend this method to a controller that settles in a minimal time in a region around the desired setpoint. Analogously, Grieder and Morari [2003], Besselmann, Lofberg, and Morari [2009] discuss an explicit MPC approach which drives the controlled system to an invariant feasible set in a minimal number of steps as a means of complexity reduction of the explicit MPC search space. These controllers are developed for a linear system and a linear parameter varying system respectively. They define for their application the invariant feasible set as a set where a linear controller respects all system and input constraints and keeps the controlled system state inside this set. However, they do not impose constraints on the settling behavior inside this invariant set. Note that for all these applications, the reachable state space is limited due to complexity reasons.

1.2.5 General review on model predictive control for mechatronic applications

MPC is an advanced control technique which can take the system constraints directly into account by the on-line solution of an optimization problem. This control technique is originally developed for systems in the process and chemical industry. These systems are typically characterized by slow system dynamics and hence allow lower sampling rates and more time to solve the optimization problem. Since the introduction of MPC , much research has been dedicated to theoretical issues regarding MPC control such as stability and robustness. It is only since the last decade that the research focus has been shifted towards the development of more efficient and faster algorithms to solve the MPC optimal control problem in real-time. However, the practical application of MPC to mechatronic systems with sampling frequencies above hundred hertz is still mostly nonexistent. Moreover, the number of decision variables is typically limited to around five to ten. Also, time-optimal control, i.e. explicitly minimizing the settling time of the system as an objective function, is not yet extensively tackled within the MPC framework.

1.3 Motivation

This section motivates and poses the main concept of this thesis: time-optimal system motion by means of controllers developed within the MPC framework.

For many mechatronic applications as e.g. overhead cranes, wafersteppers, pick-and-place machines, production machines, ..., minimization of settling time is the most important control objective. By reducing the movement time, the throughput and hence rendability of these systems can be increased. In the linear control community much work has been performed on the development of controllers which are near optimal with respect to the settling time. However, in order to generate controllers which are optimal with respect to settling time for any possible reference step, the system constraints have to be taken into account explicitly. A disadvantage of the linear controllers is that they generally cannot account for these constraints. Therefore, they are often either too conservative or they violate the actuator constraints for time critical applications.

MPC on the other hand is a control approach which determines the optimal system input by the solution of an optimization problem each sampling time. In this optimization problem, the system constraints can be taken into account explicitly. However, so far these controllers are mostly developed for and applied to non-linear systems with slow dynamics in e.g. the process industry. For these applications, long sampling periods in the seconds or minute range are sufficient and hence more time is available for the solution of the MPC optimization problem. Nowadays, MPC algorithms are also being developed for systems with fast system dynamics. In order to reduce the on-line computational load, the attainable horizons are still limited however. Moreover, time-optimality has not yet been studied extensively and the application of these controllers to realistic mechatronic setups is nearly inexistent.

Hence, this analysis of the current state-of-the-art leads to the following design hypotheses for the application of time-optimal controllers to mechatronic systems:

First, is it possible to formulate time-optimality within the MPC framework, i.e. is it possible to develop controllers which realize time-optimal behavior for all reference steps without being too conservative or without violating system constraints?

Second, is it possible to formulate the resulting optimal control problem such that it is computationally feasible, i.e. can the underlying control problem structure be exploited such that this problem can be solved for systems with sampling frequencies up to the kilohertz range?

Third, is it possible to formulate the MPC problems with long enough prediction horizons and sufficient decision variables such that they are practically relevant for mechatronic systems, i.e. can their values be increased by a factor three to five in comparison with the current-state-of-the-art for fast normal MPC applications?

Fourth, is it possible to prove that the developed controllers are behaving as desired, i.e. can it be proven that these controllers converge to the desired setpoint?

Fifth, is is possible to show that these controllers are practically relevant, i.e. can the time-optimal controllers be experimentally validated on relevant mechatronic test setups?

The theoretical developments in this thesis are based on perfect system knowledge and assume that no disturbances are present and hence full state knowledge is available. For practical implementation however, the system is subject to disturbances and the sensors introduce measurement noise. Therefore, for the practical implementation, state estimators are added to the system to estimate the system state. These estimators are however not taken into account in the controller design.

1.4 Chapter-by-chapter overview and contributions

1.4.1 Chapter-by-chapter overview

First, Chapter 2 describes the development and experimental validation of a time-optimal open-loop controller called predictive prefilter. The predictive prefilter generates time-optimal reference trajectories for point-to-point motions by solving an optimization problem on-line. This predictive prefilter has been designed as a more performant replacement of traditional prefilters. The chapter describes how time-optimality is formulated within the MPC framework. Also, it is shown how robustness can be introduced in the predictive prefilter framework. The chapter ends with an experimental validation on a mass-spring-damper system of the predictive prefilter. Also, the predictive prefilter's performance is compared with linear control approaches.

Second, Chapter 3 describes the development and experimental validation of a closed-loop time-optimal controller called time-optimal MPC (TOMPC). The TOMPC controller generates a feedback signal which minimizes the settling time by solving a series of feasibility problems. The chapter gives a theoretical analysis of the TOMPC controller. The chapter describes how the optimization problem structure is exploited in order to make this feasibility search possible in real-time. Moreover, this chapter presents how the attainable range of the controller can be increased without compromising on the computational load. The chapter ends with an experimental validation of the TOMPC controller on a linear motor drive and on an overhead crane.

Third, Chapter 4 describes the application of time-optimal controllers on an industrial linear motor setup. For this industrial setup, not only time-optimality is desired but also a very high positioning accuracy in the submicrometer range. This high accuracy can not be obtained with TOMPC only due to its relative low sampling frequency. Therefore, this chapter presents two control schemes which combine TOMPC with linear feedback. Also, the chapter presents a numerical and experimental validation of both control schemes and compares their performance with industrial benchmark requirements.

Finally, Chapter 5 concludes the thesis by summarizing the main developments and indicating the most relevant theoretical and practical extensions for future work.

1.4.2 Main contributions

The main goal of this PhD thesis is to develop time-optimal MPC based controllers for mechatronic systems. In comparison with linear controllers, MPC controllers have the advantage that they can take system constraints into account directly. Therefore, MPC controllers are better suited than linear controllers to minimize settling time as this requires an activation of a maximum number of constraints. The central contributions in the development of these controllers are fourfold. (i) Time optimality has been defined in the MPC framework. (ii) The MPC optimization problem has been formulated such that they can be solved in real-time, i.e. sampling frequencies up to 250 hertz have been obtained. (iii) The problem formulation allows to use prediction horizons that are long enough for realistic motion control applications. (iv) All developed controllers have been implemented on embedded hardware and have been validated experimentally on realistic mechatronic systems. Also, a C++ implementation of a time-optimal controller has been made publicly available on-line. Within this general setting, three controllers have been developed; an open loop controller which generates time-optimal reference trajectories, a closed loop controller which generates time-optimal feedback signals and a controller which combines time-optimality with industrial requirements on submicrometer positioning accuracy.

First, the PhD thesis develops an open-loop controller, called predictive prefilter, which transforms a reference step into a reference trajectory. This reference trajectory minimizes the settling time for point-to-point motions while taking into account the actuator constraints. In order to solve the open-loop controller optimization problem in real-time, the time-optimal optimization problem is approximated by one linear optimization problem with an exponential weighting of the absolute output error. Also, the developed open-loop controller framework is extended to incorporate robustness with respect to model-plant mismatch. The developed controller has been implemented in C++ on embedded hardware. This open-loop controller has been succesfully validated numerically and experimentally on a mass-spring-damper system. These validations show that the reformulated optimization problem underlying this controller can be solved in less than ten milliseconds on embedded hardware.

Second, the predictive prefilter concept is extended to incorporate feedback such that disturbances can be accounted for. This closed loop controller is called time optimal MPC (TOMPC). The time-optimal optimization problem has been adapted to take into account the sensitiveness with respect to measurement noise. A proof of convergence to the desired setpoint of the TOMPC controller has been presented. By exploiting the structure of the optimization problem, it is possible to solve this problem in real-time. The TOMPC controller has been implemented in C++ on embedded hardware. This code is made publicly available on-line as open-source. The TOMPC controller has been validated experimentally on an overhead crane and a linear motor at sampling rates up to 250 hertz and with up to forty-five decision variables. Both time-optimal point-to-point motion experiments and disturbance rejection experiments have been performed succesfully.

Finally, the time-optimal controller approach has been adapted for and implemented on an industrial linear motor test set-up. For this application, not only time-optimality is required, but also a settling accuracy in the submicrometer range. In order to obtain this very high positioning accuracy, sampling times in the kilohertz range are required. Therefore, the time-optimal control scheme has been combined with a linear feedback controller. Convergence of the combined controller scheme has been proven theoretically. All benchmark requirements on settling time and settling accuracy have been met succesfully in experiment. Moreover, it has been shown experimentally that the control scheme is robust with respect to model-plant mismatch. The main drawback of this control scheme is the switch between the TOMPC controller and the linear controller which can cause constraint violation. Therefore, also a second control scheme which combines time-optimality with high-accuracy positioning has been presented. This control scheme has been validated numerically.

Chapter 2

Predictive prefilter

This chapter describes the development of a prefilter within the MPC framework. The developed prefilter generates a reference trajectory which minimizes the settling time for an LTI system while respecting the system constraints. First, this chapter motivates the design of the predictive prefilter and shows how time-optimality has been formulated in the MPC framework in Section 2.1. Section 2.2 and Section 2.3 show how this optimization problem can be solved in real-time. Section 2.4 discusses how robustness can be introduced in this framework. Section 2.5 validates the prefilter on a mass-spring-damper system both by simulation and experiments and compares the designed prefilter with linear prefilter approaches. This chapter is based on [Van den Broeck et al., 2010].

2.1 The predictive prefilter

The predictive prefilter is a prefilter that minimizes the settling time for point-to-point motions taking into account system constraints. A prefilter is a system that transforms a reference signal into a system input signal without incorporating feedback, as illustrated in Fig. 2.1. A prefilter is typically used for two types of applications. First, for applications where sensor feedback is not desired. Multiple reasons exist not to include extra sensors into the system: their cost, the physical impossibility to include the sensors, the speed of response of the sensors, stability issues, . . . Second, prefilters are often used in combination with feedback controllers in a two degree of freedom control structure where the feedback controller is used to reject disturbances and the prefilter for optimal reference tracking. This second case is equivalent to the first case if the closed-loop system is defined as the total system.

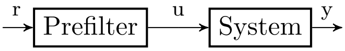

Figure 2.1: General idea of a prefilter. The prefilter filters the reference r into an input u for the system resulting in output y.

For many applications as e.g. cranes, pick-and-place machines, ..., point-to-point motions have to be performed as fast as possible, without stating requirements on the intermediate trajectory. Currently, there are two possible approaches. A first approach filters the reference step with a linear prefilter. However, these prefilters are only time-optimal for one worst case reference step and are too conservative or violate constraints for all other steps. The worst case reference step is for these applications usually defined as the largest possible reference step. A second approach computes optimized reference trajectories off-line. Hence, if the reference step is not known beforehand, the computation of the optimal trajectory introduces a delay between the step request and the actual start of control. This delay can easily be larger than the actual movement time. Moreover, this second approach can not cope with reference changes during motion. Hence, for applications where time-optimality is important and the reference steps are not known beforehand, a new type of prefilter is required which obtains time-optimal reference trajectories for all reference steps and which respects the system constraints. A prefilter which satisfies these requirements is developed in this chapter.

This section starts with the formulation of a time-optimal prefilter within the MPC framework. This prefilter is based on the input shaping design developed in [Van den Broeck et al., 2008]. The section continues with a short proof on the time-optimality of this design. The predictive prefilter has been developed in discrete-time and hence is based on a discrete-time presentation of the system.

2.1.1 Formulation of time-optimal problems in an MPC framework

The goal of the optimal control design in this thesis is to design a controller which minimizes the settling time. Conceptually, this can be formulated as the following mixed integer optimization problem $P(\bar{x}_l, y_{\text{ref},l})$ which has to be solved at every sampling time step l:

$$\min_{x,u,y,K} K \tag{2.1a}$$

$$\text{subject to: } x_0 = \bar{x}_l, \tag{2.1b}$$

$$x_{k+1} = f(x_k, u_k), \tag{2.1c}$$

$$y_k = h(x_k, u_k), \tag{2.1d}$$

$$0 \leq g(x_k, u_k) \text{ for } k = 0 \ldots K, \tag{2.1e}$$

$$x_K = f(x_K, u_K), \tag{2.1f}$$

$$y_K \equiv y_{\text{ref},l}. \tag{2.1g}$$

In this problem formulation, K is the prediction horizon length and x_k, u_k, y_k are the system state, system input and system output at time step k in the prediction horizon respectively. Constraints (2.1c)–(2.1d) describe the system dynamics which for a given input completely determine the system behavior if the initial state is known. This initial state is obtained by system simulation and the simulated value is imposed by the value $\bar{x}_l$ in constraint (2.1b). Constraint (2.1e) represents all constraints on the inputs, outputs and states. These constraints allow to take the input constraints of the system into account during optimization. Constraints (2.1f)–(2.1g) are new in comparison with traditional MPC problem (1.2). They impose that the system has to arrive without residual vibrations at the desired setpoint y_{ref} at time K. Constraint (2.1f) imposes that the system is in an equilibrium position at time step K, and (2.1g) that the equilibrium is at the correct position. The MPC controller minimizes as an objective function the time K required to arrive at this desired state (2.1a). By solving this optimization problem at every sampling time, a controller is designed that generates a reference trajectory which minimizes the settling time of the system.

2.1.2 Design of the predictive prefilter

If the considered system is linear and the imposed constraints on inputs, outputs and states are linear, problem (2.1) reduces to a special variant of a mixed integer linear problem:

$$\min_{x,u,y,K} \quad K \tag{2.2a}$$

$$\text{subject to: } x_0 = \bar{x}_l, \tag{2.2b}$$

$$x_{k+1} = Ax_k + Bu_k, \tag{2.2c}$$

$$y_k = Cx_k + Du_k, \tag{2.2d}$$

$$f \leq Hx_k + Gu_k \text{ for } k = 0 \dots K, \tag{2.2e}$$

$$x_K = Ax_K + Bu_K, \tag{2.2f}$$

$$y_K \equiv y_{\text{ref},l}, \tag{2.2g}$$

where A,B,C and D represent the system dynamics matrices. The endpoint-constraints (2.2f)–(2.2g) can be replaced by at least two equivalent formulations:

- A single constraint on the end-state:

  $$x_K = \bar{x}, \tag{2.3}$$

  possibly in combination with a constraint on the last input(s) to avoid a non-unique solution:

  $$u_K = \bar{u}, \tag{2.4}$$

  where $\bar{x}$ and $\bar{u}$ correspond to the equilibrium state and equilibrium input of the system in the required end-point position respectively. They are computed beforehand by solving a set of equality constraints similar to (2.2f)–(2.2g).

- Constraints only on the output, and not on the state:

  $$y_k = y_{\text{ref},l}, \; k = K, \dots, K+n-1, \tag{2.5}$$

  with n the number of states in the system model. This equivalence can, under some mild conditions, easily be proven for linear systems as has been shown in Section 3.2.2.

Optimization problem (2.2) can hence either be solved directly by a mixed integer solver, by an approximate solver (see Section 2.2) or by a sequence of linear feasibility problems (see Chapter 3).

2.1.3 Time optimality

Theorem 1. *For any single reference step input, the predictive prefilter generates the fastest possible reference trajectory.*

Proof. The solution of problem $P(\bar{x}_l, y_{\mathrm{ref},l})$ for a single reference step results in a time-optimal trajectory, i.e. the shortest possible move-time considering the system constraints. Because the optimizer and state predictor use the same system model, the principle of optimality of subarcs [Bellman, 1957] applies. Hence, the optimal input trajectory obtained in the first discrete time step will stay optimal throughout the whole motion. ☐

Corollary 1. *If the system is subject to a setpoint change, the predictive prefilter will be at least as fast as a linear prefilter.*

When a conventional linear prefilter is compared with the predictive prefilter, there are two possible scenarios:

- The linear prefilter is designed for this reference trajectory and hence is time-optimal; i.e. no prefilter can be faster without violating constraints. Because of its construction, the predictive prefilter will be as fast as this linear prefilter
- The linear prefilter is not designed for this reference trajectory and hence is not time-optimal; i.e. the prefilter will either not activate the constraints or will violate these constraints. The predictive prefilter will avoid both by construction. It will activate the constraints when possible, but will never violate them. Hence, the predictive prefilter generates a faster reference trajectory.

2.2 Solution strategy: L_1 penalty

The first solution approach developed within this thesis does not solve directly optimization problem (2.2) but solves a slightly different problem. By using a weighted sum of l_∞-norms of the tracking error:

$$\min \sum_{k=0}^{N} \|y_k - y_{\mathrm{ref}}\|_\infty c^k, \;\; c > 1, \tag{2.6}$$

instead of the objective function (2.2a) and endpoint constraints (2.2f)–(2.2g), similar output results can be obtained. These similar outputs can be obtained by solving at every sampling time only one linear optimization problem with a fixed prediction horizon length N instead of solving the mixed integer problem. The resulting linear optimization problem is given by:

$$\min_{x,u,y,s} \sum_{k=0}^{N} s_k c^k,\ c > 1, \tag{2.7a}$$

$$\text{subject to: } x_0 = \bar{x}_l, \tag{2.7b}$$

$$x_{k+1} = Ax_k + Bu_k, \tag{2.7c}$$

$$y_k = Cx_k + Du_k, \tag{2.7d}$$

$$-s_k \leq y_k^{[i]} - y_{\text{ref}}^{[i]} \leq s_k \text{ for all outputs } i \tag{2.7e}$$

$$f \leq Hx_k + Gu_k \text{ for } k = 0 \ldots N, \tag{2.7f}$$

where s_k are dummy variables necessary to cast the l_∞ norm into an LP. The combination of the sum of l_∞-norms and an exponential weighting results in an efficient removal of the small residuals, i.e. the residual vibrations. To that end and to get meaningful results, the different outputs have to be scaled to the same range. By using an exponential weighting c^k with $c > 1$, the normal exponential decay of the residual vibrations due to damping is compensated for, and hence the time-optimal behavior is enforced at the computational cost of only one linear optimization problem instead of multiple feasibility problems. This result is in analogy with [T.-W. Yoon and Clarke, 1993]. T.-W. Yoon and Clarke [1993] state that with an exponential weighting of the absolute output error the modulus of the closed loop discrete poles can be made arbitrarily small for unconstrained systems. This reformulation should be handled with care though. As presented in [Wang, 2001], optimal control problems become very bad conditioned if c is too large and the prediction horizon length is long. They even show that in order to keep well conditioned optimization problems, c should have a value smaller than one, which would off course destroy the time-optimal output behavior. This effectively limits the range of attainable prediction horizons. For the real-time solution, it is also important that the considered time-horizon is not too long as the computation time of the solution algorithms scales at least linearly but usually worse with the horizon length. In order to make a good choice of the prediction horizon, first a classical input shaper can be designed and the length of this classical input shaping prefilter can be used as length of the prefilter. A comparable choice is to formulate optimization problem (2.7) with an optimization horizon length equal to half a period of the slowest eigenfrequency of the system. Note that with formulation (2.7) the system does not have to reach the desired setpoint at the end of the horizon N. This allows to keep the prediction horizon relatively small.

This solution strategy has been compared with a true time-optimal controller for the test case presented in the Section 2.5. These controllers have been compared for a large variety of desired displacements and they resulted in almost identical solutions, with a maximum relative output difference of 0.4 %.

2.3 Implementation

Optimization problem (2.7) is solved with an active set strategy. In order to use this strategy, the optimization problem has first been condensed, i.e. the system states x and the equality constraints describing the system dynamics (2.7c)–(2.7d) are eliminated from the optimization problem and the optimization problem is defined only in the initial state x_0 and inputs u. Moreover, the problem is also redefined in Δu. After condensing, the output is given by:

$$Y = Y_x x_0 + Y_u u_{\mathrm{p}} + Y_{\Delta u} \Delta U, \tag{2.8}$$

where u_{p} is the input at the previous time step $l-1$ and ΔU is the vector containing all future Δ inputs. The matrices Y_x, Y_u and $Y_{\Delta u}$ are defined as:

$$Y_x = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}, Y_u = \begin{bmatrix} D \\ D+CB \\ D+CB+CAB \\ \vdots \\ D+CB+CAB+\ldots+CA^{N-1}B \end{bmatrix} \tag{2.9}$$

$$Y_{\Delta u} = \begin{bmatrix} D & 0 & \ldots & 0 \\ D+CB & D & \ldots & 0 \\ D+CB+CAB & D+CB & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ D+CB+\ldots+CA^{N-1}B & D+CB+\ldots+CA^{N-2}B & \ldots & D \end{bmatrix} \tag{2.10}$$

with A, B, C and D the discrete system dynamics matrices and N the fixed horizon length.

The resulting condensed optimization problem is solved by the on-line active set solver qpOASES [Ferreau et al., 2008]. Therefore and because the technique developed in Chapter 3 relies heavily on the principles underlying active set solvers, this paragraph gives a short introduction to active set methods (see also e.g. [Gill et al., 1984, Bartlett and Biegler, 2006]). They are a general method for solving convex quadratic optimization problems:

$$\min_{w} \; w^T H w + g^T w \tag{2.11a}$$

$$\text{s.t.} \quad Aw \leq b, \tag{2.11b}$$

with w the vector of optimization variables, (2.11a) the objective function, (2.11b) the constraints under which the objective function is optimized and H a positive definite matrix. Introducing dual variables λ, the Karush-Kuhn-Tucker (KKT) conditions for optimality can be formulated as stated in [Nocedal and Wright, 2006]:

$$Hx^\star + g - A'_{\mathbb{A}}\lambda^\star_{\mathbb{A}} = 0, \tag{2.12a}$$

$$A_{\mathbb{A}}x^\star = b_{\mathbb{A}}, \tag{2.12b}$$

$$A_{\mathbb{I}}x^\star \leq b_{\mathbb{I}} \tag{2.12c}$$

$$\lambda_{\mathbb{A}} \geq 0, \tag{2.12d}$$

$$\lambda_{\mathbb{I}} = 0. \tag{2.12e}$$

where the active set $\mathbb{A}$ is the set of all active inequality-constraints, i.e. the inequalities which constrain the solution and hence become equalities in the optimum of the optimization problem. $A_{\mathbb{A}}$ and $b_{\mathbb{A}}$ indicate respectively the rows of A and elements of b corresponding to these active constraints. $A_{\mathbb{I}}$ and $b_{\mathbb{I}}$ are analogously the rows of A and elements of b corresponding to the inactive constraints.

In order to solve optimization problem (2.11), an active set method makes a guess of the active set in the optimum, and based on this active set, optimizes subsequently the corresponding equality constrained QP by solving the following problem:

$$\begin{bmatrix} H & A^T_{\mathbb{A}} \\ A_{\mathbb{A}} & 0 \end{bmatrix} \begin{bmatrix} x \\ -\lambda_{\mathbb{A}} \end{bmatrix} = -\begin{bmatrix} g \\ b_{\mathbb{A}} \end{bmatrix}. \tag{2.13}$$

The solution of optimization problem (2.13) requires a matrix inversion of the lefthand side matrix which is implemented using matrix factorization. The resulting optimal solution of equality constrained problem (2.13) can have two possible outcomes:

- The KKT-conditions (2.12) for the full optimization problem (2.11) are not satisfied and hence a constraint has to be added to or removed from the active set.
- The KKT-conditions (2.12) for the full optimization problem (2.11) are satisfied and hence the optimal solution $x^\star$ is found.

Active set methods solve optimization problem (2.11) by adding and removing constraints to/from the active set $A_{\mathbb{A}}$ until the solution of (2.13) satisfies the KKT-conditions (2.12). They are fast and efficient since these operations of adding and removing constraints do not change the matrix $A_{\mathbb{A}}$ and its factorization much, such that subsequent problems (2.13) can be solved efficiently.

Assuming that both the model-plant mismatch and the disturbances are limited, the MPC optimization problem to be solved at every time step l does not differ much from the problem at the previous time step $l-1$. Moreover, if all QP matrices stay constant and only the gradient and constraint vectors change, the solution procedure can be hotstarted from the solution at the previous time step by reusing its factorization and performing a linear homotopy towards the new optimal solution. This technique is implemented in the on-line active set solver qpOASES [Ferreau, 2007].

2.4 Robustness

Because the system dynamics are never perfectly known, the basic predictive prefilter framework is extended to include robustness with respect to model-plant mismatch. In analogy with the Extra Insensitive input shaping prefilter [Singhose et al., 1994], robustness is introduced by solving optimization problem (2.7) not only for the nominal system but for a set of J systems. Hence, the resulting optimization problem takes the l_∞ norm over the outputs of these J systems instead of only the nominal system:

$$\min_{x,u,y,s} \sum_{k=0}^{N} s_k c^k, \;\; c > 1, \tag{2.14a}$$

$$\text{subject to: } x_0 = \bar{x}_l, \tag{2.14b}$$

$$x_{k+1,j} = A_j x_{k,j} + B_j u_k, \tag{2.14c}$$

$$y_{k,j} = C_j x_{k,j} + D_j u_k, \tag{2.14d}$$

$$-s_k \leq y_{k,j}^{[i]} - y_{\text{ref}}^{[i]} \leq s_k \text{ for all outputs } i \tag{2.14e}$$

$$f \leq H_j x_{k,j} + G_i u_{k,j} \text{ for } k = 0 \ldots N \text{ and } j = 0 \ldots J, \tag{2.14f}$$

where A_j,B_j C_j and D_j are the system dynamics matrices for the J perturbed systems. These perturbed systems do not need to have the same dimensions. The perturbed systems must be chosen based on the expected model uncertainty. First an analysis of the expected system uncertainty range has to be made. Then, the number of systems J and the parameters of these systems can be determined such that they span the system uncertainty range as good as possible. As a sensitivity curve analogous to Fig. 1.2 is expected, these extra systems are best chosen to have parameters which are in the middle of the expected uncertainty range. This robustness approach is illustrated in Paragraph 2.5.4. It should be noted that although the introduction of these extra system dynamics does not increase the number of condensed decision variables, it does increase the number of constraints and therefore increases the computational load. Therefore, the number of systems J which can be added is limited. Although this approach does not give a guaranteed limit on the residual vibrations, in practice a good reduction is obtained.

2.5 Validation

This section discusses the numerical and experimental validation of the developed predictive prefilter. First the experimental test setup is presented in Paragraph 2.5.1. The numerical validation in Paragraph 2.5.2 and 2.5.3 is based on a linear model of this test setup and clearly shows the advantages of the predictive prefilter in comparison with linear prefilters as input shaping. Paragraph 2.5.4 presents the experimental validation of the developed prefilter both for the nominal and robust case.

2.5.1 Test setup

Both for the numerical and experimental validation the same test setup has been used. This test setup can be considered as a two-DOF mass-spring-damper system. Fig. 2.2 shows a picture and a schematic drawing of the setup.

The system is excited with a hydraulic piston. The position of this piston is indicated by $p(t)$, which is measured by an LVDT position sensor. This piston operates at a maximum pressure of 140 bar and is controlled through a 2/4 valve using a 214 rad/s bandwidth PID position controller. The reference of this position controller is the system input [V], where 1 V corresponds to a desired displacement of 10 cm. The position of the upper mass $x_1(t)$ [cm] has been chosen as the system output. This system can be modeled by a fifth-order continuous-time state space model. The parameters from this model are determined based on frequency response function measurements that are obtained from a multisine excitation of the system with a frequency content between 0.1 Hz and 10 Hz [Pintelon and Schoukens, 2001]. In order to apply the developed framework, this model is transformed to discrete time with a sample period $T_s = 10\,\mathrm{ms}$. The identified model contains two pairs of complex conjugated poles originating from the two flexible modes of this system,

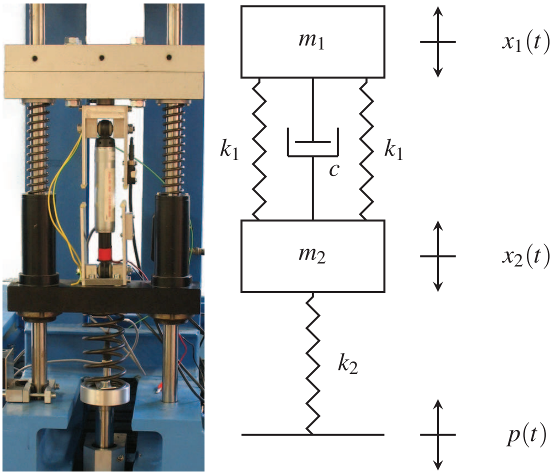

Figure 2.2: Picture and schematic drawing of the mass-spring-damper test setup.

and one real pole that is introduced by the band limited piston position controller. The poles and their damping are summarized in Table 2.1. This table shows that the complex conjugated pole pairs are only very lightly damped. Therefore an uncontrolled or badly controlled actuation of this system results in long lasting residual vibrations and hence a high settling time. The system model contains no zeros and has a DC gain of 10 cm/V. As the fastest pole lies at 34 Hz and the vibration poles are even slower, a sampling rate of 100 Hz is sufficient. The input of this system is limited to 1 V, and the output is limited to a displacement of 10 cm. These constraints on the input and output determine after condensing and with a slight abuse of notation the matrix G and vector f in expression (2.7f) as:

$$f = \begin{bmatrix} y_{\min} - Y_x x_0 - Y_u u_{\mathrm{p}} \\ -y_{\max} + Y_x x_0 + Y_u u_{\mathrm{p}} \\ u_{\min} - u_{\mathrm{p}} \\ -u_{\max} + u_{\mathrm{p}} \end{bmatrix}, G = \begin{bmatrix} Y_{\Delta u} \\ -Y_{\Delta u} \\ L_N \\ -L_N \end{bmatrix}, \tag{2.15}$$

with L_N a lower triangular matrix of ones of dimension N by N and $u_{\max}$, $u_{\min}$, $y_{\max}$ and $y_{\min}$ the upper and lower bounds on respectively the input and the output of the system. In the development of the predictive prefilter, it is assumed that the system starts at rest, i.e. the starting state is the zero state.

Table 2.1: Poles of the fifth order system

frequency [rad/s]	damping [-]
$\omega_0 = 2.6205 \times 2\pi$	$\zeta_0 = 0.157\%$
$\omega_1 = 7.7926 \times 2\pi$	$\zeta_1 = 0.293\%$
1 real pole at $34 \times 2\pi$	/

2.5.2 Benchmark test

First, the predictive prefilter has been compared numerically with a classical prefilter for a benchmark problem. The classical prefilter is designed for a maximal reference step of 10 cm. The application of this reference step without preshaping results in an overshoot of 98 % and a 5 %-settling time of 2 s. The classical prefilter has been designed following the linear optimization procedure developed in [Van den Broeck et al., 2008]. In this procedure, negative impulses are allowed as in the predictive prefilter in order to make a fair comparison on settling time. In the linear prefilter design input constraints are imposed such that no saturation takes place for the maximal reference step of 10 cm. Fig. 2.3 shows this reference (black dashed line) and the system output, obtained after applying the prefiltered reference both with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). This figure shows that the predictive prefilter can reproduce the classical input shaping prefilters for the maximal allowable reference step. Fig. 2.4 shows this more clearly by showing the outputs of the input shaping prefilter (grey solid line) and the predictive prefilter (black solid line), which are the inputs of the system, for this reference. The small difference between these filter outputs can be attributed to the fact that some small optimization freedom is left at the time-optimal solution as the real optimal settling time can only be approximated from above in a discrete time setting.

2.5.3 Validation and comparison with traditional prefilters by simulation

This paragraph shows the advantages of the predictive prefilter compared to a linear prefilter when references which are different from the maximal reference step are requested. In a first test the desired setpoint change is not a maximal step, but a sequence of smaller steps. Fig. 2.5 shows the desired setpoints for the system (black dashed line) and the resulting output behavior obtained after filtering these steps with a classic input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). This figure clearly shows the superior behavior of the predictive prefilter. Reductions of the settling time up to 30 % have been obtained. Fig. 2.6 shows the respective inputs of the system, i.e. with input shaping (grey solid line) and predictive prefiltering (black solid line). This shows that the reduction

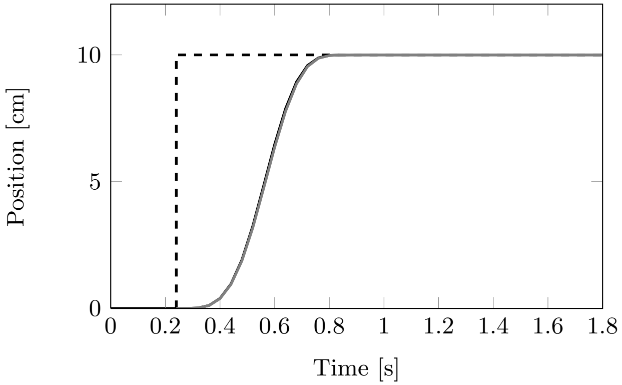

Figure 2.3: Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line), for a full step reference (black dashed line). Note that both lines coincide.

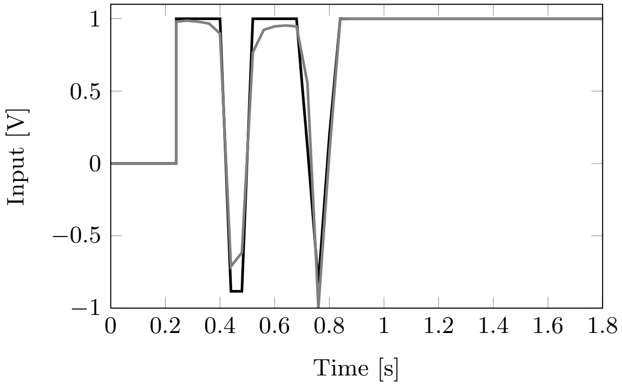

Figure 2.4: Input of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line) for a worst case scenario reference step.

in settling time is due to the more efficient use of the available input range. The classic prefilter scales its input for the maximal step with the desired step, while the predictive prefilter makes full use of the available actuator possibilities. This allows faster reaction times and hence shorter settling times.

The next example illustrates the superior settling time performance more clearly. Fig. 2.7 shows the system output behavior if a new reference set point is requested before the previous step is fully executed. Both the output with the predictive prefilter (black solid line) and with a classical input shaping prefilter (grey solid line) are shown for a reference trajectory of three steps (black dashed line).

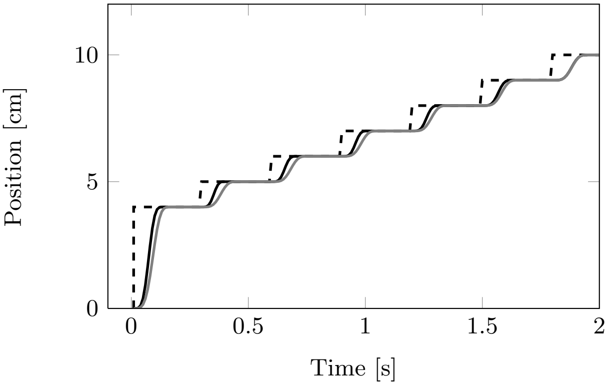

Figure 2.5: Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). The reference is not one big step, but a series of smaller steps (black dashed line).

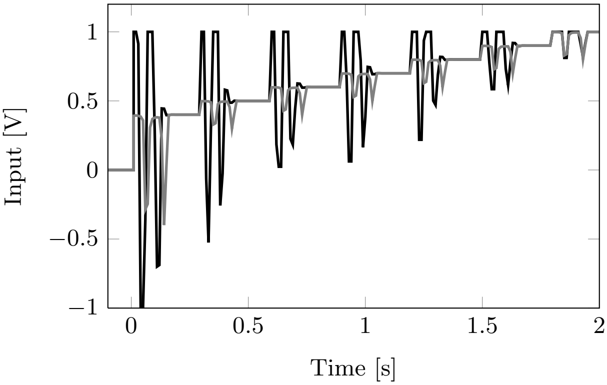

Figure 2.6: Input of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line) for a series of step references different from the worst case scenario reference step.

Fig. 2.8 presents a last simulation example which shows the better performance of the new prefilter. This example requests a reference trajectory (black dashed line) with a step up from 0 cm to 10 cm at time 0 s followed by a step down to 3 cm at time 0.1 s. A traditional input shaping prefilter with negative impulses generates a filtered signal which saturates the actuators of the system. This results in an unexpected system behavior with lots of residual vibrations (grey solid line). The predictive prefilter (black solid line) can take the system constraints into account when determining the ideal input in real-time. Therefore, this prefilter generates a reference which does not saturate the system actuators, and hence is also for

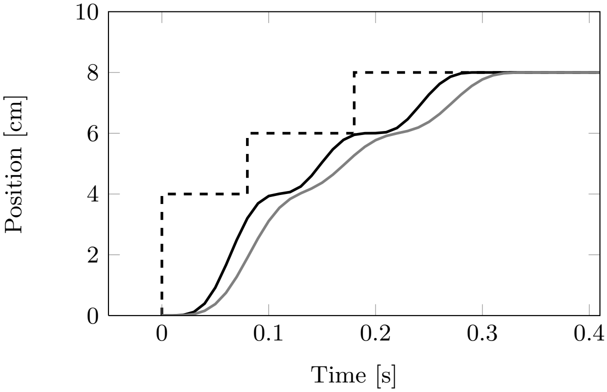

Figure 2.7: Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). The reference is a series of three steps (black dashed line), where a new setpoint is requested before the previous setpoint is attained.

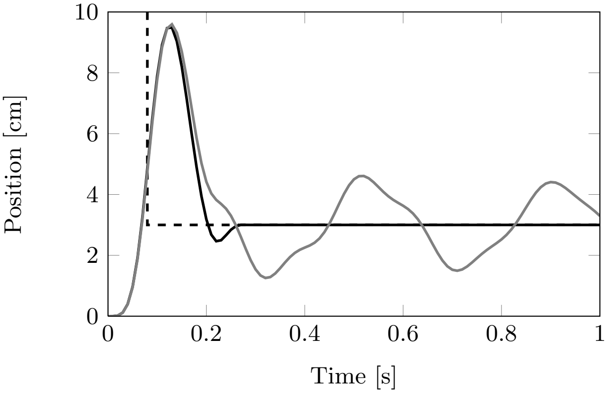

Figure 2.8: Output of the system with a classical input shaping prefilter (grey solid line) and a predictive prefilter (black solid line). The reference is a series of two steps (black dashed line), where the second step is reversed compared with the first step.

this case more efficient. Note that if the classical input shaping prefilter would be designed with only positive impulses, this saturation does not occur. However, the settling time for the experiments shown in Figs. 2.3, 2.5 and 2.7 would be much higher and hence the superiority of the predictive prefilter for these cases would even be bigger.

2.5.4 Validation and comparison with traditional prefilters by experiment

After extensive numerical validation, the predictive prefilter design also has been validated experimentally. The considered test setup is the mass-spring-damper setup presented in Paragraph 2.5.1. The prefilter is implemented in Simulink and runs on a dSPACE DS1103 controller board containing a 1 GHz processor with 90 MB RAM.

The considered optimization problem for the experiments is formulated with a horizon of $K = 15$ time steps and $c = 2$ in objective function (2.7a). The input is limited to ± 1 V. In addition, an input slew rate constraint of 10 V/s has been imposed because of oil flow limitations of the hydraulic setup. This constraint of 10 V/s has been determined based on the technical specifications of the hydraulic system and has been verified experimentally. An additional overshoot constraint of 5 % is imposed. Hence, the resulting optimization problem (2.7) contains for this application 30 variables (15 decision variables and 15 dummy variables for the infinity norm) and 180 constraints (30 for the input, 30 for the slew rate of the input, 30 for the l_∞ norm, 30 for the overshoot and 30 dummy constraints on the dummy variables which are required for the implementation). The algorithm presented in Section 2.3 is implemented in C++ in S-functions within the Simulink environment. The required maximal and mean CPU time for the solution of each optimization problem on the dSPACE platform is respectively 2.9 ms and 0.5 ms. The maximal computation time of 2.9 ms is required when a new reference step is requested. This maximal CPU time is still safely below the 10 ms sampling time, making real-time application of the method clearly possible.

Fig. 2.9 shows the system response, i.e. the position $x_1(t)$, when a step reference (black dashed line) is applied filtered both with the designed predictive prefilter (black solid line) and with a traditional input shaping prefilter (grey solid line). The traditional input shaping prefilter has been recomputed to take into account the slew rate constraints. This experiment confirms the 30 % settling time reduction obtained by numerical validation. The small peaks on the position signal are measurement noise. Fig. 2.10 shows the system input generated by filtering the reference step with the predictive prefilter. The difference between this input profile and the profile observed in Figs. 2.4 and 2.6 is due to the slew rate constraints which are the true limiting factor during movement.

A second experiment validates the robust prefilter on a perturbed system. The perturbed system is generated by reducing the first eigenfrequency of the nominal system with 20 %. Fig. 2.11 shows the output in simulation of the nominal system (black solid line) and of the perturbed system (grey solid line) when a reference step (black dashed line) is applied through a non-robust prefilter to the plant. This figure clearly shows the necessity of robustness for these levels of model-plant

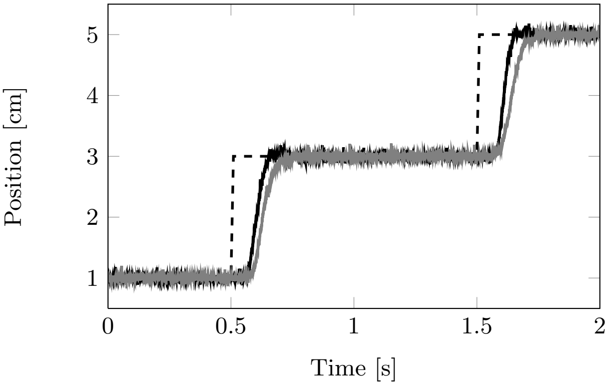

Figure 2.9: Experimental validation of the predictive input shaping prefilter design: reference motion (black dashed line), and system response obtained by prefiltering with a classical input shaping prefilter (grey solid line) and with the predictive prefilter (black solid line).

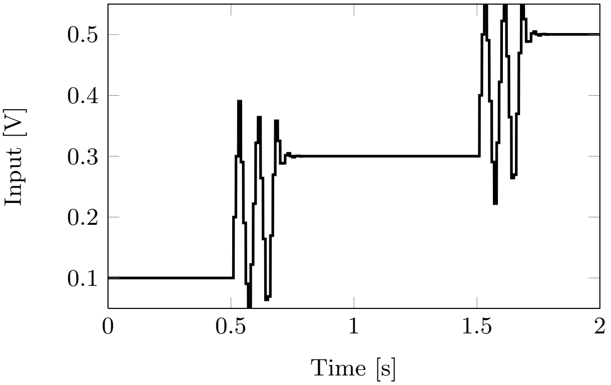

Figure 2.10: Optimal experimental input to the system computed by the predictive prefilter.

mismatch.

In order to reduce the sensitiveness to model-plant mismatch also a robust prefilter has been developed. First, masses are added to m_1 of the setup (see Fig. 2.2) such that the first eigenfrequency of the system is reduced by 20 %. In the design of the robust predictive prefilter, it was known that a relatively large system uncertainty was expected. Therefore, a robust prefilter has been developed based on $J = 3$ systems. The two extra systems in optimization problem (2.14a) have a change in eigenfrequency of 10 %. Fig. 2.12 shows the experimental behavior (black solid line) of the perturbed test setup. Fig. 2.13 shows the corresponding system input. These

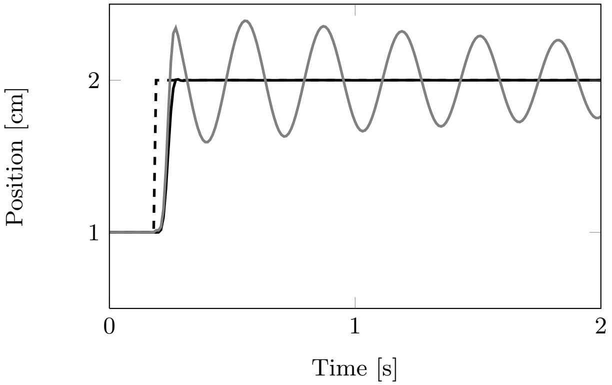

Figure 2.11: Output of a perturbed system (grey solid line) and the nominal system (black solid line) for a reference step (black dashed line) filtered with a non-robust prefilter.

figures show that the resulting robust response is still free of residual vibrations, however at the cost of an increase in settling time of 0.32 s in comparison with the non-robust prefilter approach. As the maximum computation time for this robust implementation is 7.6 ms, which is safely below the sampling time of 10 ms, this implementation was also implementable on the embedded hardware. Table 3.1 gives an overview of the mean and maximum computation times for the non-robust and robust implementation (if $J = 3$ systems are considered). This shows that the maximal computation time for both implementations is well below the sampling period of 10 ms. If the number of systems J in the robust implementation is increased further, the computation time increases. Numerical experiments show that considering $J = 5$ system yields a maximum computation time of 8 ms. Hence, the resulting prefilter would still be computationally feasible for system which have a sampling period of 10 ms. It should be noted that for the considered active set solution approach, the computation time mainly depends on the number of system inputs and outputs, and is almost independent of the number of system states. The number of constraints on the other hand does have a big influence on the computational load. For the considered case study here, the number of inputs is equal to one and the number of outputs is equal to the number of systems J considered in the robust implementation.

2.6 Conclusion

The predictive prefilter discussed in this chapter generates a time-optimal point-to-point motion reference trajectory for an LTI system by solving in real-time

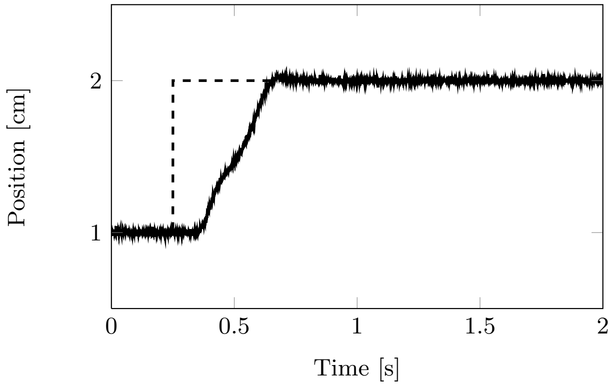

Figure 2.12: Robust design experiment: a reference step (black dashed line) is applied to a perturbed system with a 20 % change in eigenfrequency. This reference is robustly prefiltered and the resulting response (black solid line) is still free of vibrations.

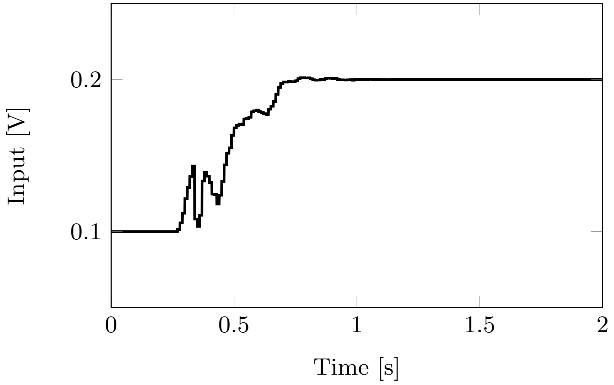

Figure 2.13: Optimal experimental input to the system computed by the robust predictive prefilter.

an optimization problem taking into account system constraints. The predictive prefilter realizes a non-linear mapping between the sequence of setpoints and the generated reference inputs. This is the main difference with classical prefilters which are LTI systems. The real-time optimization approach yields time-optimal reference inputs for any point-to-point motion while respecting system constraints and it can cope with setpoint changes during motion. These advantages comes at the cost of more expensive real-time control hardware however.

The main contributions of this chapter are as follows. First time-optimality has been defined within the MPC framework and it has been proven that the prefilter defined

Table 2.2: Required on-line computation time for the robust and non-robust predictive prefilter. A maximal computation time of 10 ms is allowable because of sampling period constraints.

problem	max comp. time [ms]	mean comp. time [ms]
non robust	2.9	0.5
robust	7.6	1.7

within this framework generates reference trajectories that minimize the settling time given the system model and system constraints. Then, robustness with respect to model-plant mismatch is introduced into this framework. In order to reduce the computational load, the true time-optimal optimization problem has been approximated by a linear optimization problem which minimizes an exponential weighting of the output error. An advantage of this formulation is that only one optimization problem has to be solved every sampling time. A disadvantage of this approach is the very bad condition of the resulting optimization problem due to this exponential weighting which prohibits the use of long optimization horizons. The developed controller has been implemented in C++ . The controller has been validated both numerically and experimentally on a mass-spring-damper system. The simulations show that in comparison with traditional linear prefilters, gains of 30 % in settling time can be achieved easily. The experiments show that the developed controller can be implemented on embedded hardware and still satisfies the real-time requirements, i.e. the underlying optimization problems can be solved in ten milliseconds or less. Moreover it has been shown experimentally that the robustified predictive prefilter can cope with model-plant mismatch.

Chapter 3

Time optimal MPC

This chapter discusses the development of a closed-loop MPC controller which minimizes the settling time on a reference step. First, Section 3.1 motivates the development of a full feedback time-optimal MPC (TOMPC) controller. This section continues by showing how the TOMPC optimization problem can be exactly reformulated within the MPC framework. Second, Section 3.2 discusses how the structure of the reformulated optimization problem underlying the controller can be exploited to make a solution in real-time feasible for mechatronic systems. Finally, Section 3.3 shows the experimental validation of the developed control strategy on a linear motor drive and an overhead crane. This section also compares the performance of the TOMPC controller with linear controllers and traditional MPC controllers. This chapter is based on [Van den Broeck et al., 2011].

3.1 Time optimal MPC controller design

In the previous chapter, an open-loop time-optimal controller for mechatronic systems has been developed. This controller can either be applied to an open-loop system as validated in the previous chapter or as a reference generator to a closed-loop system. A disadvantage of the open-loop controller is that disturbances can either not be accounted for when no extra feedback controller is applied or that the prefilter can still generate a reference trajectory which saturates the system actuators due to the effect of unknown disturbances and modeling errors. This saturation can be avoided for limited disturbances, however at a cost of introducing conservativeness [Gilbert and Kolmanovsky, 1999, 2002]. Therefore, a closed-loop controller which directly controls the system inputs has been developed. In the design of this closed-loop controller, the sensitivity to measurement noise close to the setpoint has been taken into account as a high sensitivity and fast reaction on a reference step is highly desired but this aggressive reaction is undesired on sensor measurement noise.

3.1.1 TOMPC problem formulation

In Section 2.1.1, optimization problem (2.1) shows how time-optimality can be described within the MPC framework. In the previous chapter, this optimization problem is solved approximately with an exponentially weighted l_1 norm based objective function. This chapter solves the problem exactly by a feasibility search. To account for measurement noise, the optimization problem which is solved at every sampling time is extended to:

$$\min_{x,u,y,K} K \tag{3.1a}$$

$$\text{subject to: } x_0 = \bar{x}_l, \tag{3.1b}$$

$$x_{k+1} = f(x_k, u_k), \tag{3.1c}$$

$$y_k = h(x_k, u_k), \tag{3.1d}$$

$$0 \leq g(x_k, u_k) \text{ for } k = 0 \ldots K, \tag{3.1e}$$

$$x_K = f(x_K, u_K), \tag{3.1f}$$

$$y_K \equiv y_{\text{ref},l}, \tag{3.1g}$$

$$K \geq N_{\min}, \tag{3.1h}$$

where x_k, u_k and y_k are the system states, system input and system output at time k in the prediction horizon. Constraints (3.1c)–(3.1d) impose the system dynamics and (3.1e) imposes the constraints on inputs, outputs and states. Constraints (3.1f)–(3.1g) require the system to be at rest at the desired setpoint at the end of the optimization horizon K. Constraint (3.1b) defines the initial state of the optimization problem which for the closed loop control problem is determined by on-line state measurement or state estimation. Constraint (3.1h) puts a lower limit on the optimization horizon K. This allows to increase insensitivity to measurement noise close to the setpoint, i.e. if the system is at most $N_{\min}$ steps away from the setpoint, the controller does not want to minimize K further and only requires the system to arrive in $N_{\min}$ steps at the setpoint. Hence, the controller has optimization freedom left to react less aggressively on measurement noise. This constraint transforms the time-optimal solution slightly. However, as shown in

Section 3.1.3, the corresponding degradation in settling time is negligible if chosen correctly. In order to solve this mixed integer optimization problem, the TOMPC controller is designed as follows by a two level optimization problem.

'Problem A': First, a standard MPC optimization problem is defined as 'Problem A', and denoted by $P_A(\bar{x}_l, N)$, to stress its dependence on the initial value $\bar{x}_l$ and the horizon length N:

$$V_A^\star(\bar{x}_l, N) = \min_{\substack{x_0,\dots,x_N \\ u_0,\dots,u_{N-1}}} \sum_{k=0}^{N-1} \|u_k - u_{\text{ref}}\|_R^2 + \|x_k - x_{\text{ref}}\|_Q^2, \tag{3.2a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_l, \tag{3.2b}$$

$$x_{k+1} = f(x_k, u_k), \tag{3.2c}$$

$$g(x_k, u_k) \geqslant 0 \quad k \in [0, N-1], \tag{3.2d}$$

$$x_N = x_{\text{ref}}, \tag{3.2e}$$

where endpoint constraint (3.2e) requires the system to be at the reference state at the end of the prediction horizon N. The matrices Q and R in objective function $V_A^\star$ (3.2a) are required to be positive definite. This objective function $V_A^\star$ is extended to $V_A^\star = \infty$ if $P_A(\bar{x}_l, N)$ is infeasible. The physical meaning of infeasibility of P_A is that the system can not settle at the reference point x_{ref} in N time steps while respecting all constraints. This allows to define an admissible set $\mathbb{X}(N)$:

$$\mathbb{X}(N) = \begin{cases} \{\bar{x}_l | P_A(\bar{x}_l, N) \quad \text{is} \quad \text{feasible}\} \\ \{\bar{x}_l | V_A^\star(\bar{x}_l, N) \quad \text{is} \quad \text{finite}\}. \end{cases} \tag{3.3}$$

Hence, $\mathbb{X}(N)$ is the set of system states from which the setpoint can be reached in at most N time steps, while respecting the system dynamics (3.2c) and all system constraints (3.2d).

'Problem B': Second, the mixed integer optimization problem 'Problem B', which is denoted by $P_B(\bar{x}_l)$, is defined as follows:

$$V_B^\star(\bar{x}_l) = \min_{N \in \mathbb{N}} N \tag{3.4a}$$

$$\text{s.t.} \quad N \geq N_{\min}, \tag{3.4b}$$

$$N \leq N_{\max}, \tag{3.4c}$$

$$\bar{x}_l \in \mathbb{X}(N), \tag{3.4d}$$

where N is the required settling time, $N_{\min}$ is a minimal bound on N and $N_{\max}$ is the maximal optimization horizon. Note that the level sets of $V_B^\star$ in $\bar{x}_l$ space are plateaus of height N:

$$V_B^\star(\bar{x}_l) = \begin{cases} \infty & \text{if} \quad V_A^\star(\bar{x}_l, N_{\max}) = \infty \\ N & \text{if} \quad V_A^\star(\bar{x}_l, N) < \infty \quad \& \quad V_A^\star(\bar{x}_l, N-1) = \infty \\ & \quad \& \quad N \geq N_{\min} \quad \& \quad N \leq N_{\max} \\ N_{\min} & \text{if} \quad V_A^\star(\bar{x}_l, N_{\min}) < \infty \end{cases} \tag{3.5}$$

Fig. 3.1 illustrates the level sets for a system with two states.

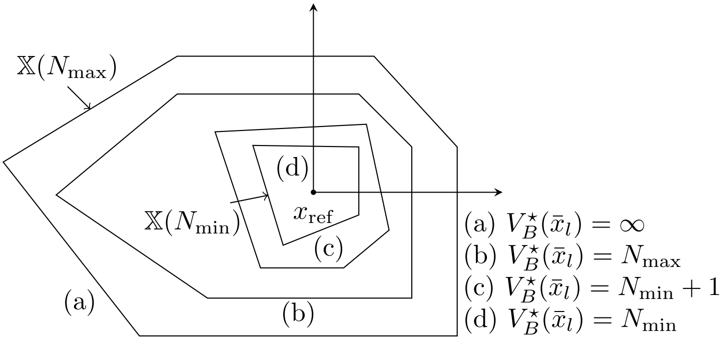
$\mathbb{X}(N_{\max})$
$\mathbb{X}(N_{\min})$
(d)
x_{ref}
(c)
(a)
(b)
(a) $V_B^\star(\bar{x}_l) = \infty$
(b) $V_B^\star(\bar{x}_l) = N_{\max}$
(c) $V_B^\star(\bar{x}_l) = N_{\min} + 1$
(d) $V_B^\star(\bar{x}_l) = N_{\min}$

Figure 3.1: Visualization of level sets of $V_B^\star$ in $\bar{x}_l$ space

At each time step, problem P_B which minimizes the settling time N is optimized. As problem P_A is underlying P_B, also a traditional MPC problem with endpoint constraints is optimized if optimization freedom is left, i.e. if $N = N_{\min}$ or if multiple time optimal solutions exist. This two-layer optimization problem therefore produces the desired functionality: the system approaches the setpoint as fast as possible while respecting the constraints on inputs and states, and close to the setpoint the system reacts as traditional MPC with endpoint constraints on measurement noise and disturbances. The choice of the weights Q and R in (3.2) can therefore be tuned to obtain the desired regulating behavior in the neighborhood of the setpoint.

3.1.2 Proof of asymptotic convergence

In order to have a useful TOMPC controller, it is important to have a guaranteed convergence towards the desired settling point, even when $N_{\min}$ is different from 0:

Lemma 1. *If for some $N \in \mathbb{N}$ holds $\bar{x}_l \in \mathbb{X}(N)$, then also $\bar{x}_l \in \mathbb{X}(N+1)$.*

Proof. Because $\bar{x}_l \in \mathbb{X}(N)$, a feasible solution $(\tilde{x}^l, \tilde{u}^l)$ of $P_A(\bar{x}_l, N)$ exists [1]. Then, by construction and by endpoint constraint (3.2e), $([\tilde{x}^{lT}, x_{\text{ref}}^T]^T, [\tilde{u}^{lT}, u_{\text{ref}}^T]^T)$ satisfies all constraints of $P_A(\bar{x}_l, N+1)$ and is hence a feasible solution, which implies that $\bar{x}_l \in \mathbb{X}(N+1)$. □

Lemma 2. *If* $(\tilde{x}^l, \tilde{u}^l) = (x_0, x_1, \dots, x_N, u_0, u_1, \dots, u_{N-1})$ *solves* $P_A(\bar{x}_l, N)$*, with* $x_0 = \bar{x}_l$*, then* $(x_1, \dots, x_N, u_1, \dots, u_{N-1})$ *solves* $P_A(x_1, N-1)$*.*

Proof. This lemma is a corollary of Bellman's principle of optimality of subarcs [Bellman, 1957]. □

Lemma 3. *For an unperturbed system without model-plant mismatch and assuming uncorrupted plant measurements, feasibility at time* $l = 0$ *guarantees feasibility at each time* $l > 0$*.*

Proof. By assumption, a feasible solution of $P_A(\bar{x}_0, N)$ exists at $l = 0$, which is denoted by $S_0 = (\tilde{x}^0, \tilde{u}^0)$. The closed loop control action at the first time step is hence the first element of $\tilde{u}_0$, indicated as $\tilde{u}_0^0$. At time $l = 1$, a new TOMPC optimization problem is formulated with initial state $\bar{x}_1$, where $\bar{x}_1 = \tilde{x}_1^0$ because of the assumption of no model-plant mismatch and no disturbances. Hence, if $N > N_{\min}$, a feasible solution of $P_A(\bar{x}_1, N-1)$ exists because of Lemma 2. If $N = N_{\min}$, then because of endpoint constraint (3.2e), $([\tilde{x}_{(1\dots N)}^{0T}, x_{\text{ref}}^T]^T, [\tilde{u}_{(1\dots N-1)}^{0T}, u_{\text{ref}}^T]^T)$ is a feasible point of $P_A(\bar{x}_1, N_{\min})$. The lemma follows by induction. □

Theorem 2. *For each* $\bar{x}_0 \in \mathbb{X}(N_{\max})$*,* TOMPC *generates a closed-loop response for an undisturbed system without model-plant mismatch, asymptotically attracted by* x_{ref}*, with* $N_{\min} \geq 1$*; i.e.* $\bar{x}_l \to x_{ref}$ *when* $l \to \infty$*.*

Proof. First, consider $\bar{x}_l \notin \mathbb{X}(N_{\min}+1)$. It is clear from Lemma 3, that if a feasible solution with $V_B^\star(\bar{x}_l) = N$ exists at time step l, there exists a feasible solution at time step $l+1$ with $V_B^\star(\bar{x}_l) = N - 1$. Hence, every time step, the system will be driven closer to $\mathbb{X}(N_{\min})$. Second, consider $\bar{x}_l \in \mathbb{X}(N_{\min})$. The TOMPC reduces then to a traditional MPC with endpoint constraints. It can easily be proven that this local controller is therefore asymptotically attracted to x_{ref}, see e.g. [Keerthi and Gilbert, 1988]. □

3.1.3 Choice of $N_{\min}$

The formulation of the TOMPC optimization problem requires the choice of $N_{\min}$. By introducing this parameter, the 'true' time-optimal behavior is not obtained

---

[1] $(\tilde{x}^l, \tilde{u}^l)$ indicates a feasible solution set $(x_0, \dots, x_N, u_0, \dots, u_{N-1})$ of $P_A(\bar{x}_l, N)$. $\tilde{x}_i^l$ and $\tilde{u}_i^l$ indicate the $i-1$th element of these sets

anymore. However, in order to guarantee unconstrained solvability, N will normally always be bigger than n/n_u with n the number of system states and n_u the number of system inputs [Kailath, 1980]. In order to make the controller less sensitive to measurement noise and to avoid too aggressive behavior when the setpoint is reached, a higher value of N_{min} is recommended, e.g. two to three times larger. However, N_{min} should not be chosen too high, as this would destroy the time-optimal behavior. Note also that the minimum number of steps is only attainable for unconstrained controllable linear systems. Fig. 3.2 illustrates the degradation of the 1 %-settling time as a function of N_{min} for the second order system of Chapter 4. This figure illustrates that if the value of N_{min} is well chosen, the degradation in settling time is limited. The noise insensitivity is clearly illustrated in Section 3.3 on the linear motor drive.

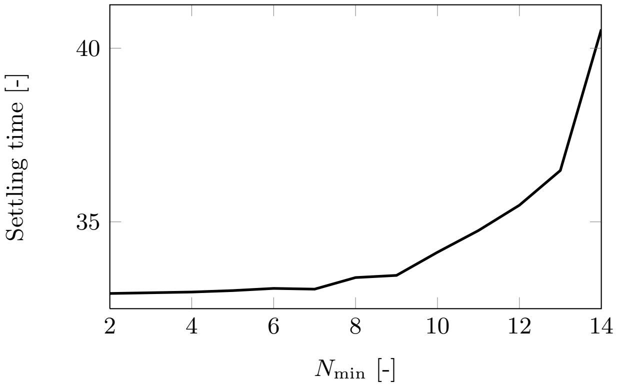

Figure 3.2: The 1 % settling time expressed as number of sample times for the second order system presented in Chapter 4 as a function of the lower bound on the settling time N_{min}.

It should be noted that time-optimal behavior can be approximated in a traditional MPC optimization problem by tuning the weights Q and R very aggressively, i.e. the weight on the output error Q much larger than on the input cost R. However, even by putting $R = 0$, the same behavior can not be obtained. Other approaches based on l_1 norm objective functions as in Chapter 2 can better approximate a time-optimal controller. However, these controller would also react very aggressively on noise, a property which can be avoided by the lower bound on the feasibility search. A disadvantage of the formulation developed in this chapter is the employment of endpoint constraints which require the controller to see the setpoint at the end of the prediction horizon.

3.2 Real-time Implementation

This section discusses the real-time implementation of the two level TOMPC optimization problem defined in Section 3.1.1. First, the TOMPC optimization problem is further analyzed. Then, it is explained how the structure of the problem is exploited in order to yield a TOMPC implementation that is sufficiently efficient for mechatronic applications. Finally, it is discussed how the feasibility region of the controller can be extended, without compromising on the computational speed.

3.2.1 Problem formulation

In general, optimization problem (3.2) is a non-convex problem. However, if the system dynamics (3.2c) and all constraints (3.2d) are linear, as is the case for the considered class of LTI mechatronic systems, this problem reduces to a convex QP:

$$V_A^\star(\bar{x}_l, N) = \min_{\substack{x_0,\dots,x_{N_{\max}-1} \\ u_0,\dots,u_{N_{\max}}}} \sum_{k=0}^{N_{\max}-1} \|u_k - u_{\text{ref}}\|_R^2 + \|x_k - x_{\text{ref}}\|_Q^2, \tag{3.6a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_l, \tag{3.6b}$$

$$x_{k+1} = Ax_k + Bu_k, \tag{3.6c}$$

$$e \leqslant Hx_k + Gu_k \quad k \in [0, N_{\max} - 1], \tag{3.6d}$$

$$x_k = x_{\text{ref}} \quad k = N, \tag{3.6e}$$

$$u_k = u_{\text{ref}} \quad k \in [N, N_{\max} - 1]. \tag{3.6f}$$

Note that in this formulation, extra input variables $(u_N, \dots, u_{N_{\max}-1})$ and state variables $(x_{N+1}, \dots, x_{N_{\max}})$ are added. Also, an extra constraint (3.6f) on the extra input variables is added in order to keep this optimization problem (3.6) equivalent to optimization problem (3.2). As this results in optimization problems of constant size during the feasibility search, this allows the use of on-line active set methods as discussed in Section 2.3. Henceforth, the thesis will implement, exploit the problem structure and validate this type of controller for an LTI system. It should be noted that this kind of controller can be applied for non-linear systems as well and the asymptotic convergence proof will still be valid. The solution in real-time will then be even more challenging however. For this linear system, the defining equation for the reference state is:

$$x_{\text{ref}} = Ax_{\text{ref}} + Bu_{\text{ref}}, \tag{3.7a}$$

$$y_{\text{ref}} = Cx_{\text{ref}}. \tag{3.7b}$$

Depending on the value of $\bar{x}_l$, two different optimization problems are to be solved. If $\bar{x}_l \in \mathbb{X}(N_{\min})$, the total optimization problem reduces to a regular MPC problem with endpoint constraints, which is under the above assumptions a QP. This QP can be solved efficiently using an on-line active set method as qpOASES [Ferreau, 2007–2009]. If $\bar{x}_l \notin \mathbb{X}(N_{\min})$ the resulting optimization problem $P_B(\bar{x}_l)$ is a mixed integer problem in one integer variable N. This mixed integer problem can be solved by a series of feasibility problems $P_A(\bar{x}_l, N)$, i.e. a series of QPs thanks to the property of quasi-convexity, see Lemma 1, of problem P_B. Algorithm 1 summarizes this procedure.

Algorithm 1 Optimization procedure

```
input: $\bar{x}_l$
output: $u^\star$ or error ‘infeasible problem’
start with initial guess for $N$
solve QP problem $P_A(\bar{x}_l, N)$ (3.6)
if $P_A(\bar{x}_l, N)$ feasible then
  while $P_A(\bar{x}_l, N)$ was feasible do
    store $u^\star = u_0(\bar{x}_l, N)$
    $N = N - 1$
    if $N \geq N_{\min}$ then
      solve QP-problem $P_A(\bar{x}_l, N)$
    else
      break
    end if
  end while
else
  while $P_A(\bar{x}_l, N)$ was infeasible do
    $N = N + 1$
    if $N \leq N_{\max}$ then
      solve QP-problem $P_A(\bar{x}_l, N)$
    else
      error ‘infeasible problem’
      break
    end if
  end while
  store $u^\star = u_0(\bar{x}, N)$
end if
```

3.2.2 Efficient reformulations

The following two paragraphs discuss how to speed up the solution method. First, by hotstarting the active set method on a given time step based on the solution from the previous time step, which is an application of the techniques developed in [Ferreau et al., 2008]. Second, by reformulating endpoint constraints (3.6e)–(3.6f) to optimize N with a minimum number of active set changes.

Transition over time using an online active set strategy

When the controller propagates from time step l to $l+1$, two different scenarios are possible:

- The setpoint does not change: Because the system state does not change much during one time step, neither will the optimal solution S. Hence, the number of active set changes compared with this previous solution is limited. By hotstarting the active set method from this previous solution S_l, the new solution S_{l+1} can be found efficiently. As an (almost) maximum number of constraints will be active in the optimal solution if $N > N_{\min}$, an other effective hotstart strategy does not start from the previous solution, but initializes the homotopy with a shifted solution. The initial solution guess has the previous optimal value of N reduced by one. This second strategy is implemented in the controllers used in Section 3.3 to validate TOMPC on the linear motor drive and overhead crane.
- The setpoint changes: Because the desired setpoint changes, the number of active set changes compared with the previous solution S_l can be high. Hence, the optimization will not reuse the previous solution. However, this problem can be hotstarted by making an educated guess of the optimal value of N based on simulations which are performed beforehand. These optimal guesses of N for a set of reference step lengths are stored in a table which is available for the TOMPC algorithm. When a new reference step is applied, the closest upper approximation is selected from the table as an initial guess for N.

Optimization of N with a minimum number of active set changes

Once $N = N_{\min}$, the QP-solver only has to solve one regular QP (3.6). However, if $N > N_{\min}$, a series of optimization problems has to be solved within one sampling time step, until the optimal value $N^\star$ is found. In these optimization problems, N is increased or decreased by one, each time resulting in at least $2n + n_u$ active set changes as the deactivation of constraint (3.6e) for $k = N$ requires n active set changes, the subsequent activation of (3.6e) for $k = N \pm 1$ requires another n active set changes, and the deactivation or activation of constraint (3.6f) requires analogously n_u active set changes. Therefore a different approach is proposed resulting in less active set changes for each transition from $k = N$ to $k = N \pm 1$. First, one single output which yields an observable state space model is introduced. This output can be the controlled output, as in Section 3.3 for the overhead crane and linear motor, but this is not strictly necessary. If more outputs would be needed to obtain observability, a straightforward modification of the approach described below has to be applied. Second, (3.6) is extended by introducing extra variables $(u_{N_{\max}}, \dots, u_{N_{\max}+n-2})$ and $(x_{N_{\max}+1}, \dots, x_{N_{\max}+n-1})$ and constraints (3.6e)–(3.6f) are replaced by:

$$y_{N+k} = y_{\text{ref}} \quad \text{for} \quad k = 0, \dots, n-1, \tag{3.8a}$$

$$u_{N+k} = u_{\text{ref}} \quad \text{for} \quad k = 0, \dots, N_{\max} - N - 1, \tag{3.8b}$$

where $y_{\text{ref}} = Cx_{\text{ref}}$ is the output corresponding to the steady state x_{ref}. Note that although the number of variables has been increased, all optimization problems during the feasibility search still have the same size. Lemma 4 proves that these sets of constraints are equivalent.

Lemma 4. *If the observability matrix* C_o*:*

$$C_o = \begin{bmatrix} C^T & A^T C^T \dots & (A^{n-1})^T C^T \end{bmatrix}^T \tag{3.9}$$

is of full rank, constraints (3.8) *are equivalent with* (3.6e)–(3.6f)*.*

Proof. (3.6e)–(3.6f) ⇒ (3.8): trivial.
(3.8) ⇒ (3.6e)–(3.6f): All constraints on u for the extended horizon are the same. Because of the system dynamics (3.6c), it is known:

$$x_{N+1} = Ax_N + Bu_N \tag{3.10a}$$

$$= Ax_N + Bu_{\text{ref}} \tag{3.10b}$$

$$= A(x_N - x_{\text{ref}}) + x_{\text{ref}} \tag{3.10c}$$

$$\Leftrightarrow x_{N+1} - x_{\text{ref}} = A(x_N - x_{\text{ref}}), \tag{3.10d}$$

where (3.10a) is equivalent to (3.10b) because of constraint (3.8b) and (3.10b) is equivalent to (3.10c) by definition of x_{ref} (3.7). By induction, this can be extended to:

$$x_{N+k} - x_{\text{ref}} = A^k (x_N - x_{\text{ref}}) \tag{3.11}$$

Hence, (3.11) allows to rewrite (3.8a) into:

$$C(x_N - x_{\text{ref}}) = 0 \tag{3.12a}$$

$$CA(x_N - x_{\text{ref}}) = 0 \tag{3.12b}$$

$$\vdots$$

$$CA^{N+n-1}(x_N - x_{\text{ref}}) = 0, \tag{3.12c}$$

which is equivalent to:

$$\begin{bmatrix} C' & A'C' & (A^2)'C \dots & (A^{n-1})'C' \end{bmatrix}' (x_N - x_{\text{ref}}) = 0. \tag{3.13}$$

Hence, if the observability matrix (3.9) is invertible, this equation has only one solution, and $x_N \equiv x_{\text{ref}}$. ☐

The advantage of formulation (3.8) in comparison with constraints (3.6e)–(3.6f) is that only $2 + n_u$ instead of $2n + n_u$ constraints have to be added to or removed from the active set when the value of N increases or decreases by one, which improves the numerical efficiency considerably. For example, for the first test case considered in Section 3.3 and a reference step of 10 cm, the worst case computation time (which typically occurs at the time sample when a new reference step is applied) reduces with a factor 4 by using formulation (3.8). In addition to this measure to reduce the computation time, the similarity between two consecutive problems can be exploited. Since the value of N changes by one in the series of optimization problems, the optimal solution of each problem has $2 + n_u$ known active set changes, which can be imposed directly and simultaneously [Ferreau, 2007–2009]. E.g. if N is reduced to $N - 1$, the following active set changes can be imposed: fix $u_{N-1} = u_{\text{ref}}$, fix $y_{N-1} = y_{\text{ref}}$ and relax y_{N+n-1}. By imposing these active set changes directly, and starting from the corresponding active set, the optimization procedure is accelerated further, resulting in an extra reduction of 40 % of the worst case computation time for the same test case considered above. In comparison to an interior point solver as e.g. Mosek [ApS, 2008], the worst case computation time is 10 times smaller. Table 3.1 shows the relative comparison of worst case computation times in Matlab, for the considered case using Mosek and the three discussed alternative implementations of TOMPC using qpOASES as a solver. Fig. 3.3 shows how this computation time evolves when subsequent optimization problems are solved, starting with a new reference step at time 0 s. This figure shows both the traditional effects of hotstarting and the gain in computation time thanks to the techniques developed in this section when new reference steps are requested.

Combination of the above measures to speed up the solution yields that the optimization problem underlying the TOMPC can be solved in real-time at 200 Hz and 60 Hz for respectively the linear drive system and the overhead crane as discussed in Section 3.3 and 250 Hz for the industrial test case considered in Chapter 4. It should be noted that MPC based on active set numerical solution methods have been applied with sampling frequencies in the kHz-range [Wills et al., 2005]. The sampling frequencies obtained with TOMPC are considerably lower. This is (i) because in the cases considered in this thesis, the number of decision variables and constraints is respectively approximately 4 and 10 times higher than in the case considered in [Wills et al., 2005], and (ii) because TOMPC has to solve a sequence of feasibility problems in order to obtain time-optimal behavior while Wills et al. [2005] minimizes the traditional quadratic objective function by the solution of one optimization problem.

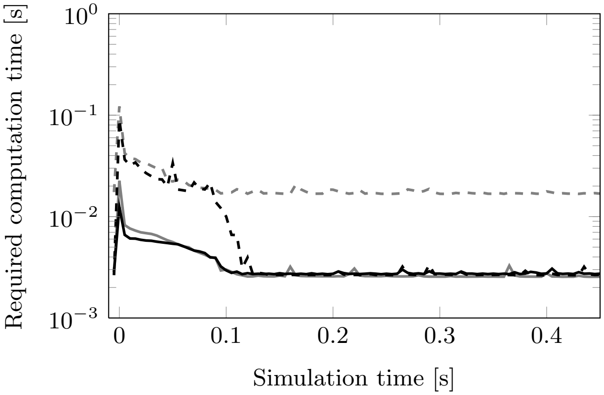

Figure 3.3: Required computation time in simulation in Matlab for the linear motor drive when a new reference step is requested at time 0 s for solution with Mosek (dashed grey line), qpOASES based on (3.6e)–(3.6f) (dashed black line), qpOASES based on (3.8a)–(3.8b) (solid grey line) and qpOASES based on (3.8a)–(3.8b) with shifting (solid black line).

Table 3.1: Relative comparison of worst case computation times in Matlab for the linear motor drive system discussed in Section 3.3 for a reference step of 10 cm, using four different solution methods.

method	maximal computation time [-]
Mosek	10.17
qpOASES based on (3.6e)–(3.6f)	8.25
qpOASES based on (3.8a)–(3.8b)	1.83
qpOASES based on (3.8a)–(3.8b) with directly imposed active set changes	1

3.2.3 Efficient extension of the prediction horizon

A disadvantage of the above described TOMPC algorithm, is the dependence of the largest possible reference step on the value of $N_{\max}$ because endpoint constraints (3.6e) and (3.6f) must be satisfied for $N \leq N_{\max}$, that is, the system must be able to reach this largest possible reference position in no more than $N_{\max}$ time steps without violating the system constraints (3.6d). $N_{\max}$ itself determines the size of the optimization problem and hence the worst case computation time of the TOMPC solution method. This worst case computation time is limited by the sampling period. Hence, the sampling period limits the largest possible reference step, which can be small for systems that require high sampling frequencies. One measure which partially alleviates this problems is using an open end of optimization problem P_B, i.e. when $N = N_{\max}$ the system is not required to be at rest at the desired setpoint. This prevents infeasibility of the total TOMPC optimization problem if the reference step is too large. In order to alleviate the step limitation more correctly, non-equidistant time steps or time gridding is applied. Time gridding allows to consider larger horizons without increasing the total number of discretization points $N_{\max}$ and hence the number of optimization variables. In the first part of the horizon up to typically $N_{\min}$, the time step corresponds to the sampling period. Thereafter, the time steps are gradually increased up to ten times the sampling time. For the overhead crane test setup of Section 3.3.2, this allows to increase the prediction horizon by a factor of 4, extending the attainable range to its maximum of 70 cm.

It should be noted that the rigorous asymptotic convergence results presented in Section 3.1.2 do not hold strictly when a non-equidistant grid is used. However, the practical closed-loop performance is nearly identical as when a fine equidistant grid is used for the same horizon length, while the computation time is significantly reduced.

An implemented version of the developed TOMPC controller in C++ is available at `http://www.kuleuven.be/optec/internal/software/44-general/139-TOMPC`. The manual of this package is presented in Appendix A.

3.3 Validation

This section validates the TOMPC concept on two realistic mechatronic test setups. The two setups available at KULeuven are respectively a linear motor drive and an overhead crane. The goal of this validation is to show that the developed TOMPC controllers are solvable in real-time on embedded hardware and that by using these controllers, time-optimal control can be obtained. Robustness with respect to model-plant mismatch has not been tested explicitly as the controllers developed in this chapter are closed loop controllers and not open loop controllers as in Chapter 2. First, in Section 3.3.1, the linear motor drive setup is presented, followed by an experimental validation of TOMPC and a comparison with both MPC and linear controllers. Section 3.3.2 discusses the application of TOMPC to an overhead crane, showing the capabilities of the TOMPC control technique to control a higher order system with a very lowly damped mode. Also, the disturbance rejection capabilities of the controller are illustrated.

3.3.1 Linear motor drive

The setup

The considered test setup is the linear motor (type LSE10G1010 of Baumuller) shown in Fig. 3.4. This type of motor is typically used in e.g. pick-and-place machines. The control input to the system is the current applied to the motor [A], and the output is the position of the linear drive [m] measured with a linear encoder with a resolution of 2 nm. The system input is limited to ±17 A due to peak current limitations, and the input slew rate is limited to ±2 A/ms because of limitations of the internal current controller. The system controllers are embedded on a SpeedGoat real-time target machine [Speedgoat, 2010] which contains a 2.13 GHz processor with 1 GB RAM. The TOMPC controllers are implemented through C++ functions in the XPC-target environment of Simulink. A discrete-time third order state space model with a sampling period of $T_s = 0.005\,\mathrm{s}$ is identified for this system, based on frequency response function (FRF) measurements that are obtained from multisine excitations with a frequency content between 0.01 Hz and 15 Hz [Pintelon and Schoukens, 2001]. This systems contains one integrator pole. Figure 3.5 shows the measured FRF (black solid line) and the FRF of the identified discrete model (grey solid line).

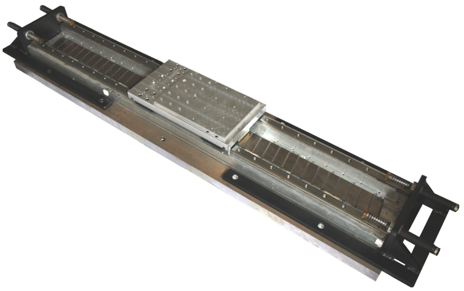

Figure 3.4: Picture of the linear motor drive test-setup in the PMA-lab at KULeuven.

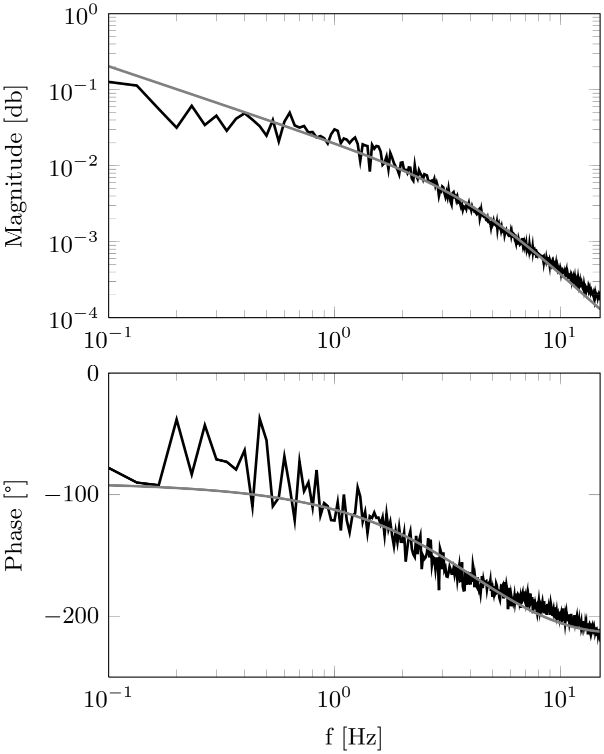

Figure 3.5: Measured (black solid line) and identified (grey solid line) FRF of the linear drive system.

Controller setup

The TOMPC and MPC controllers are implemented taking the input constraints described above into account, and in addition constraints on the overshoot and undershoot are imposed as they are necessary for the settling time constraints. The TOMPC controller is designed with a prediction horizon $N_{\max} = 45$ and weights $Q = 4\,\mathrm{m}^{-1}$ and $R = 5\,\mathrm{A}^{-1}$. The underlying optimization problem P_A contains 272 constraints (3.6d) (92 for overshoot and undershoot, and 90 for respectively the input u and the slew rate of the input Δu). It should be noted that the aggressive time optimal behavior is obtained by the shrinking horizon N and not by the choice of the weights Q and R. Hence, these weights can be selected by considering sensor noise sensitivity and disturbance regulation only, e.g. according to general tuning rules [Franklin et al., 2001], taking into account the range of the different variables, which can be normalized by applying a proper scaling. This is not the case for the traditional MPC controller, where Q and R also determine the response speed. The traditional MPC is tuned very aggressively by trial-and-error, with a much higher weight on the output error than the input cost, in order to obtain approximately time-optimal behavior (a Q-R ratio of 400 000 A/m). Since only the motor position is measurable, the state variables are determined using a state estimator. This state estimator is designed using pole placement, and has a bandwidth of approximately 85 Hz.

Comparison with linear controller

First a set of simulations is performed to illustrate the advantage of TOMPC with respect to linear feedback control, the currently dominant control approach for these systems. A desired point-to-point motion of 10 cm is considered. To make a fair comparison, the linear controller is combined with a reference trajectory instead of a step. A polynomial spline reference trajectory is optimized with respect to the time-duration of the motion for a displacement of 10 cm taking into account the requested constraints, using the convex framework developed in [Demeulenaere et al., 2009a]. This framework simultaneously optimizes the feedforward signal that yields perfect reference tracking. Fig. 3.6 compares the TOMPC output results (black solid line) and linear controller output results (grey solid line) for the point-to-point reference motion of 10 cm (black dashed line). Fig. 3.7 shows the corresponding control signals. The following conclusions can be drawn: TOMPC and the linear feedback controller in combination with the optimized feedforward yield comparable results. The small difference is due to the fact that the optimized feedforward is a spline that is continuous up to the first derivative, a condition not enforced upon the TOMPC control signal, and due to the choice of $N_{\min} > n$. The advantage of TOMPC with respect to linear feedback control becomes clear if different point-to-point motions are considered. By solving the optimization problem on-line, TOMPC will yield time-optimal behavior for any desired displacement and this without any delay between the step request and the start of control. This is in contrast with the linear controller which in order to achieve time-optimality requires a new off-line computation in order to obtain an optimized reference trajectory and corresponding feedforward for each new displacement. This off-line optimization causes a delay between the step request and the start of control which can easily be as long as the performed step. Another linear approach is to scale the reference trajectory and feedforward which is optimal for one particular reference step for all requested steps. However, this introduces conservativeness for references which are smaller than the design reference step. Moreover, for step references which are bigger than the design reference step, the system actuators will saturate and unpredictable system behavior will occur. Fig. 3.8 illustrates this for a desired displacement of 5 cm by comparing the system output obtained by control with the TOMPC controller (black solid line) and with the linear controller combined with the reference trajectory and feedforward signal (grey solid line), optimized for the 10 cm displacement but scaled down to 5 cm.

The settling time obtained with the TOMPC controller is considerably lower than the settling time obtained with the linear controller since the TOMPC controller uses the whole input range while the linear controller uses only 50 %, a direct consequence of the linear reference input scaling (see Fig. 3.9). If the same reference input is scaled up to a displacement larger than 10 cm, the actuators will saturate, yielding unpredictable, often oscillatory or unstable, behavior.

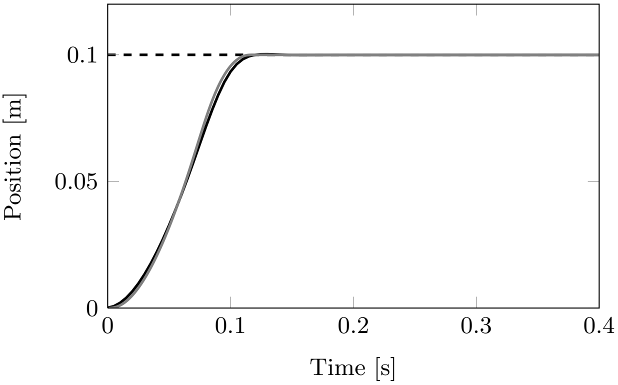

Figure 3.6: Simulated system motion for a desired displacement of 10 cm (black dashed line), obtained by using a TOMPC controller (black solid line) and by using a linear feedback controller combined with an optimized reference trajectory and feedforward (grey solid line).

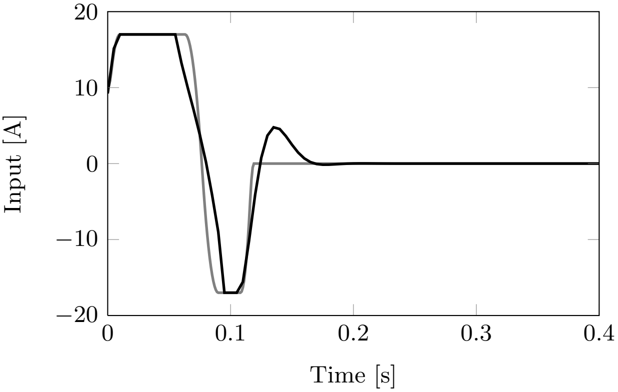

Figure 3.7: Simulated control signal for a desired displacement of 10 cm, obtained by using a TOMPC controller (black solid line) and by using a linear feedback controller combined with an optimized reference trajectory and feedforward (grey solid line).

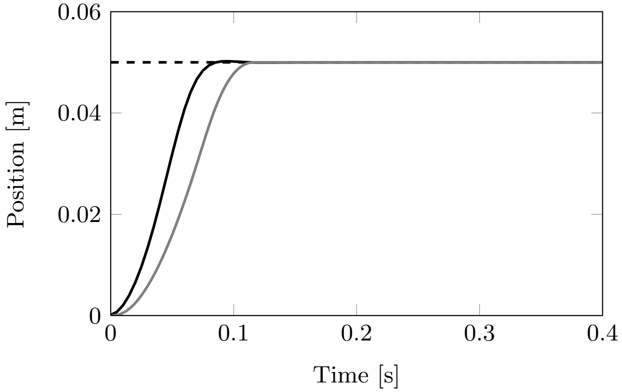

Figure 3.8: Simulated system motion for a desired displacement of 5 cm (black dashed line), obtained by using a TOMPC controller (black solid line) and by using a linear feedback controller combined with a scaled reference trajectory and feedforward (grey solid line).

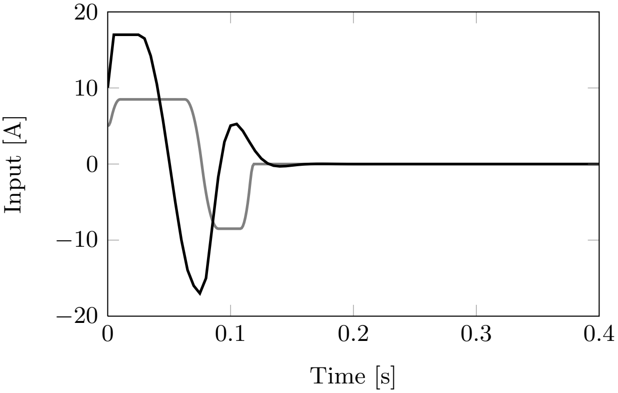

Figure 3.9: Simulated control signal for a desired displacement of 5 cm, obtained by using a TOMPC controller (black solid line) and by using a linear feedback controller combined with a scaled reference trajectory and feedforward (grey solid line).

Experimental validation and comparison with MPC on a linear motor drive

TOMPC and MPC are compared and validated experimentally on the linear motor drive system. Three sets of experiments have been performed. In a first set of experiments, TOMPC has been compared with traditional MPC for several different displacements. The resulting 1 % settling-times are summarized in Table 3.2. Fig. 3.10 shows the output motion for a reference step of 10 cm (black dashed line). Although the rise time obtained with MPC (grey solid line) is even slightly lower than with TOMPC (black solid line), the settling time with TOMPC is considerably reduced. This is more clearly shown in Fig. 3.11, which presents the output error relative to the applied step on a logarithmic scale. The remaining output error is due to model-plant mismatch and non-linear disturbances like cogging and friction. This static error can be eliminated by introducing disturbance states, see e.g. [Pannocchia and Rawlings, 2003], which is similar to introducing integrating action in linear controllers. Fig. 3.12 shows the corresponding inputs to the system. Comparable inputs are applied during the first part of motion. However when the system settles, TOMPC (black solid line) generates a different control input for the system actuators than traditional MPC (grey solid line). This similar behavior during the first part of the motion (up to 0.13 s) is because both controllers hit the input constraints during acceleration and deceleration (because of respectively the endpoint constraint and high Q-R-ratio). When this is not the case, the controllers behave differently.

Table 3.2: The 1% settling time for different steps using TOMPC and MPC

step [cm]	TOMPC [s]	MPC [s]
1	0.12	0.16
5	0.16	0.22
10	0.18	0.25
15	0.22	0.30

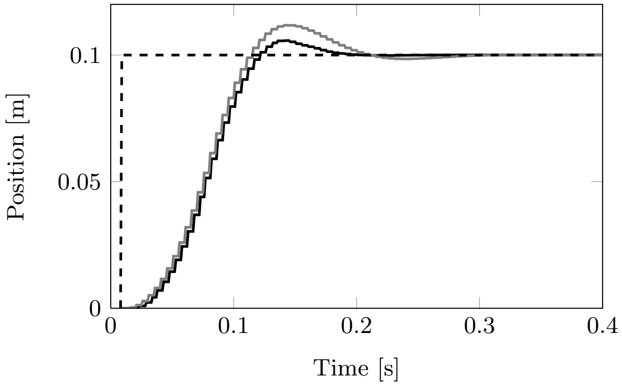

Figure 3.10: Linear drive system: Drive system motion obtained with TOMPC (black solid line) and MPC (grey solid line) for a reference step of 10 cm (black dashed line)

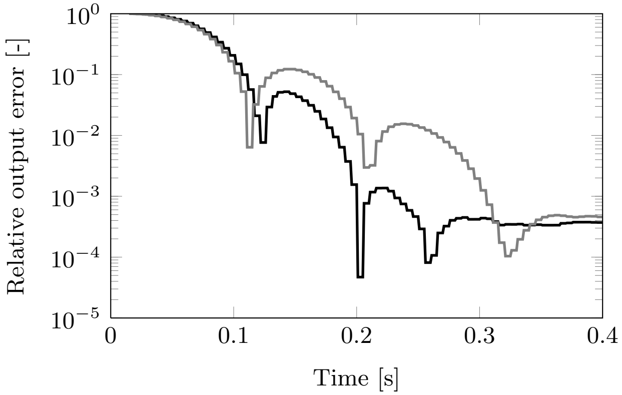

Figure 3.11: Linear drive system: Relative output error obtained by control with TOMPC (black line) and MPC (grey line) for a reference step of 10 cm

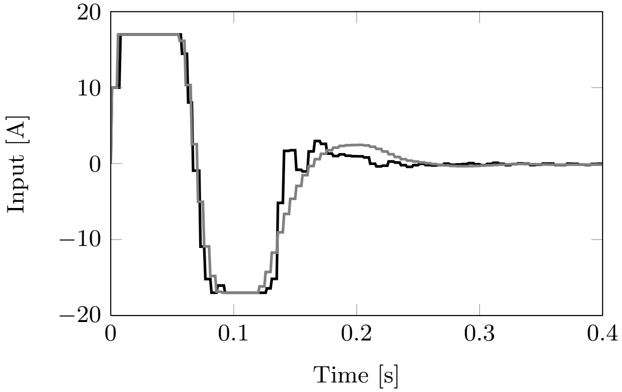

Figure 3.12: Linear drive system: Control input signal obtained by control with TOMPC (black line) and MPC (grey line) for a reference step of 10 cm

A second set of experiments shows the importance of the choice of $N_{\min}$. Due to the feedback of measurement noise and disturbances, TOMPC behaves too aggressively if $N_{\min}$ is chosen too small because the too short horizon will enforce deadbeat reactions whereas a longer horizon enables a relaxed response. Fig. 3.13 illustrates this by showing the input applied to the actuators of the system for two different choices of the minimal bound on the settling time $N_{\min}$, namely $N_{\min} = 8$ (black line) and $N_{\min} = 6$ (grey line). When a step is requested to the system at time 0 s, both controllers request the same system input and hence yield the same rise and settling time. However, close to the setpoint from time 0.2 s the controller with

$N_{\text{min}} = 8$ is much less sensitive. This effect of measurement noise and disturbances can also be reduced by decreasing the bandwidth of the state estimator, however this only reduces the effect and can not eliminate it.

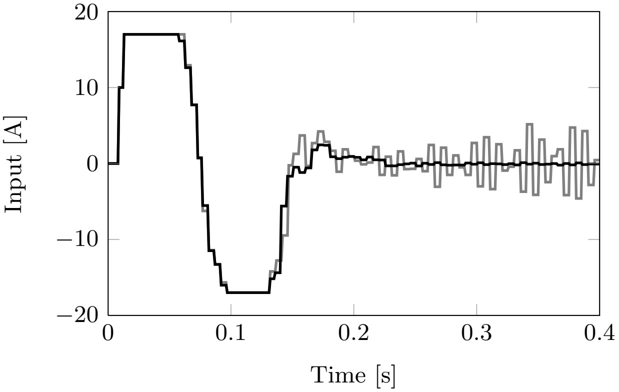

Figure 3.13: Linear drive system: Control input applied to the system by using a TOMPC controller with $N_{\text{min}} = 8$ (black line) and $N_{\text{min}} = 6$ (grey line).

A third experiment shows that although this controller is designed for completing steps with minimal settling time, it can accept a new reference step before the previous step is completed. Fig. 3.14 shows the system output (black solid line) if two steps of respectively 5 cm and 7 cm are requested (black dashed line). Fig. 3.15 shows the corresponding system input. By comparing this figure with Fig. 3.13, it can be seen that when the second reference step is requested, the input reaches its maximal value again in order to fully accelerate instead of requesting the minimal value to decelerate.

3.3.2 Overhead Crane

Test setup

The second test setup on which the TOMPC controller has been validated is the overhead crane with fixed cable length shown in Fig. 3.16. Fig. 3.17 shows a schematic representation of this system. The actuator of the system is a velocity controlled DC-motor that drives a trolley through a rack and pinion. The position of the trolley x is measured using an angular encoder mounted on the DC-motor axle, yielding a position measurement resolution of 3 µm. The swing angle θ is measured using a rotative encoder mounted on the axle to which the cable is attached, yielding an angular resolution of 0.0009°. The input to this system is a voltage u, which is a reference applied to the 25 Hz bandwidth internal velocity

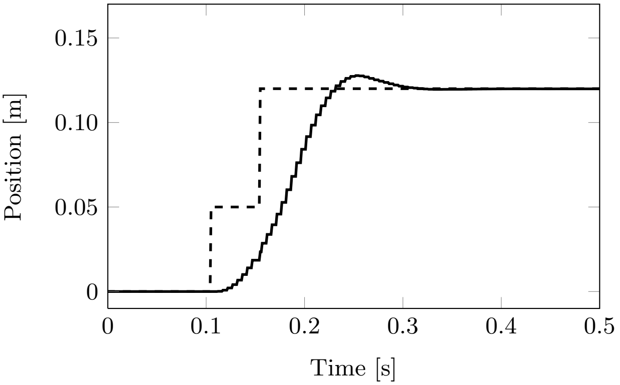

Figure 3.14: Linear drive system: Drive system motion (full line) obtained when a reference trajectory of two steps is requested (dashed line). The second step is requested before the first one is completed.

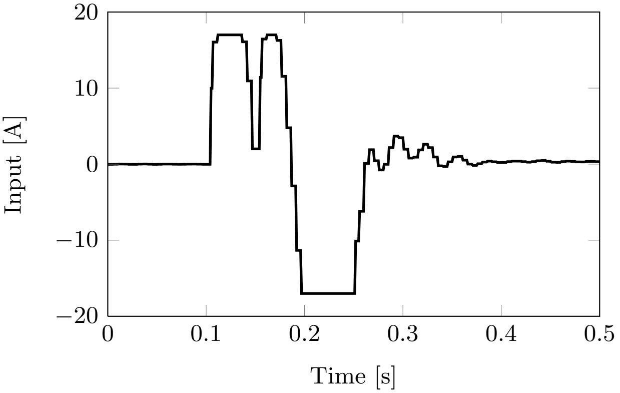

Figure 3.15: Linear drive system: Control input obtained when a reference trajectory of two steps is requested and the second step is requested before the first one is completed.

loop. The input is limited to ±1 V and the input slew rate is limited to ±6 V/s due to limitations of the motor current amplifier. The maximal range of the trolley is 70 cm. The length of the cable is fixed to 450 mm. The system controllers are embedded on a dSPACE board DS1103 which contains a 1 GHz processor with 90 MB RAM. The controllers are implemented through C++ functions in the real-time-target environment of Simulink. The system controllers are applied at a sampling frequency of 60 Hz.

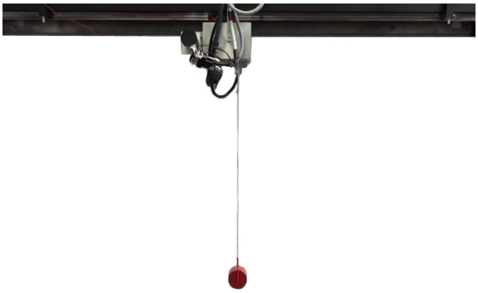

Figure 3.16: The overhead crane in the PMA-lab at KULeuven.

The relation between the input u and the position of the trolley x is modeled by a first order model for the internal velocity loop in combination with an integrator relating velocity to position:

$$\frac{X(s)}{U(s)} = \frac{K}{s(\tau s + 1)}. \tag{3.14}$$

The relation between the position of the trolley x and swing angle θ can for small values of θ be modeled as [Fliess et al., 1994]:

$$\frac{\theta(s)}{X(s)} = \frac{s^2}{Ls^2 + g}, \tag{3.15}$$

with L the length of the cable and g the gravitational acceleration. These models (3.14)–(3.15) have been combined and discretized, to yield following discrete time models which relate the input u to the position of the upper mass x and the swing angle θ:

$$\frac{X(z)}{U(z)} = \frac{b_0 z}{(z-1)(z-a_0)}, \tag{3.16a}$$

$$\frac{\theta(z)}{U(z)} = \frac{\beta_0 z(z-1)}{(z^2 + \alpha_1 z + \alpha_2)(z-a_0)}. \tag{3.16b}$$

The parameters of models (3.16a)–(3.16b) have been identified using a nonlinear least square frequency domain identification approach based on FRF measurements that are obtained from multisine excitations with a frequency content between 0.05 Hz and 5Hz [Pintelon and Schoukens, 2001]. Fig. 3.18 and Fig. 3.19 show the results of this identification, that is, a good fit between the measured FRF's (black line) and the FRF's of the two identified models (grey line). The estimated resonance frequency is 0.74 Hz which corresponds to the theoretical value $\frac{1}{2\pi}\sqrt{\frac{g}{L}}$

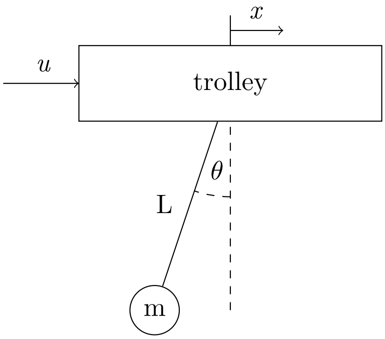

Figure 3.17: Schematic representation of the overhead crane

with $L = 450\,\mathrm{mm}$. The damping of the estimated resonance frequency is $\zeta = 0.00168$ which is extremely low.

The two identified models have been combined into a fourth order state space model of which the system dynamics matrices are given by:

$$A = \begin{bmatrix} 0.996 & 0.00435 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0.0438 & -0.0438 & 1.99 & -0.9997 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -826.1 \\ 0 \\ 35.99 \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} -0.0122 & 0 & 0 & 0 \\ 0 & 0 & 0.036 & 0 \\ -0.0122 & 0 & 0.2827 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.17}$$

where the first two states represent the position of the trolley in encodersteps at time steps l and $l-1$ and the last two states represent the swing angle of the mass in encodersteps at time steps l and $l-1$. The first output of the system is the position of the trolley x and the second output is the swing angle θ. The third output is the position of the load $y = x + \mathrm{L}\theta\frac{\pi}{180}$ in mm. This expression is valid assuming small values of θ.

State estimation

As direct measurement of the system model states results in a much too noisy state estimation, first a Luenberger observer [Luenberg] is designed. For the estimation of the states, the measurements of both the position of the trolley x and the swing angle θ are used. To explicitly take into account the higher uncertainty on the swing angle, the state estimator for the crane is developed using a Kalman filter

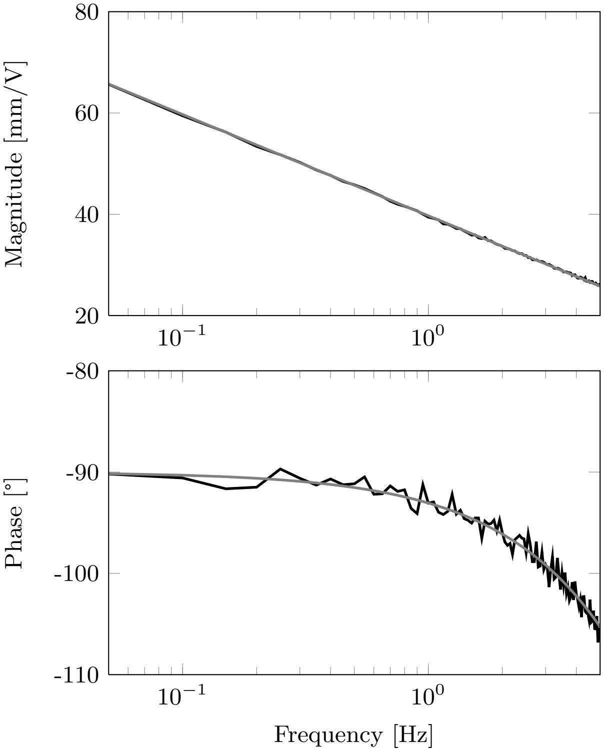

Figure 3.18: Measured FRF (black line) and FRF of the identified model (grey line) relating the input u to the position of the trolley x

[Kalman, 1960] instead of by pole placement, the technique applied for the design of the linear motor state estimator in Section 3.3.1. The best control results are obtained by using a state observer which has a process noise of 1 encoderstep, and measurement noise of respectively 1 encoderstep on the position of the trolley and 10000 encodersteps on the swing angle. The extra state introduced in the next section has a process noise of 100 V. This results in a bandwidth of 0.95 Hz. Higher observer bandwidths yield a too nervous or even unstable behavior. This maximal observer bandwidth is comparable to the maximum attainable stable bandwidth of a pure linear controller with state estimation for this setup. Note that for the linear motor drive application in Section 3.3.1 a much higher observer bandwidth and therefore a faster reaction on disturbances can be obtained.

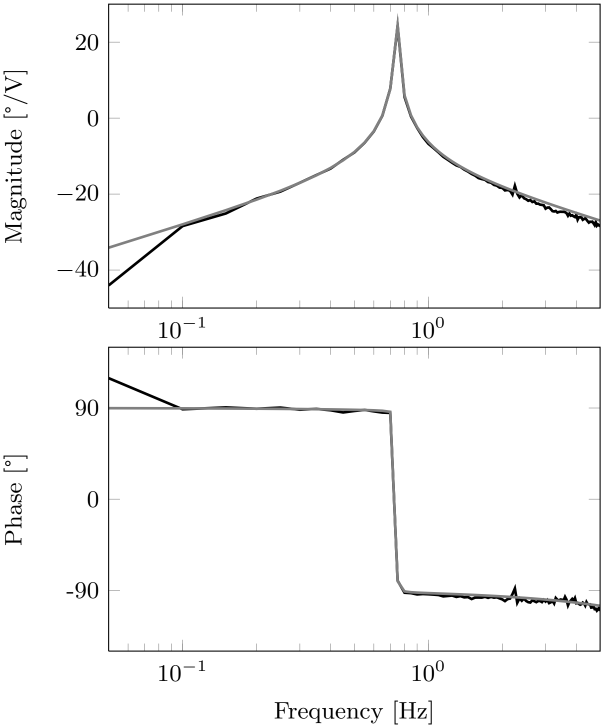

Figure 3.19: Measured FRF (black line) and FRF of the identified model (grey line) relating the input u to the swing angle θ.

TOMPC controller

For the overhead crane, a TOMPC controller has been developed taking into account the above mentioned constraints on input and input slew rate. As the solution of optimization problem (3.4) requires a large computation time, the control input is only available near the end of every sample period, and this introduces an additional sample delay. This is in contrast with a pure linear controller where the input is nearly immediately available after state estimation as the computation of the optimal input then often only requires a matrix vector multiplication. To account for this delay, the state space model is extended with one delay state, resulting in the following new state space matrices:

$$A' = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}, \quad B' = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$C' = \begin{bmatrix} C(3) & D(3) \end{bmatrix}, \quad D' = 0,$$

where $C(3)$ and $D(3)$ denote the third row of C and D in (3.17) respectively, since the output considered in endpoint constraint (3.8a) of the TOMPC implementation, is the position of the lower mass. Based on this fifth order model, the TOMPC controller has been implemented. Fig. 3.20 and 3.21 show the response of the system with TOMPC on a reference step of 10cm. Fig. 3.22 shows the corresponding input to the system. The constraints on the input amplitude are not reached. The input slew rate however, shown in Fig. 3.23, reaches the slew rate constraints during almost the whole duration of the motion. This shows that the system is working at its limits and hence indicates that the controller steers the system as fast as possible to the desired endpoint.

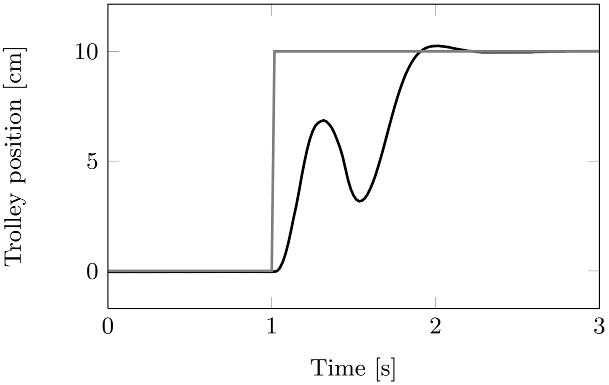

Figure 3.20: Position of the trolley (black line) controlled by a TOMPC controller for a reference step of 10 cm (grey line).

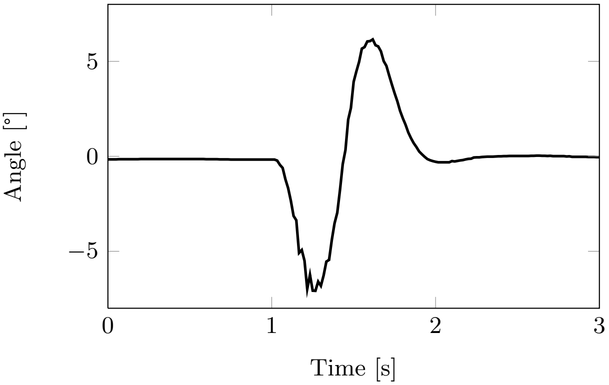

Figure 3.21: Swing angle of the TOMPC controlled system for a reference step of 10 cm.

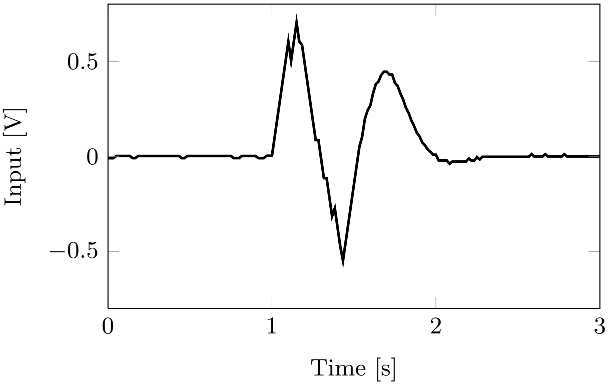

Figure 3.22: Input applied to the overhead crane for a requested step of 10 cm controlled by a TOMPC controller.

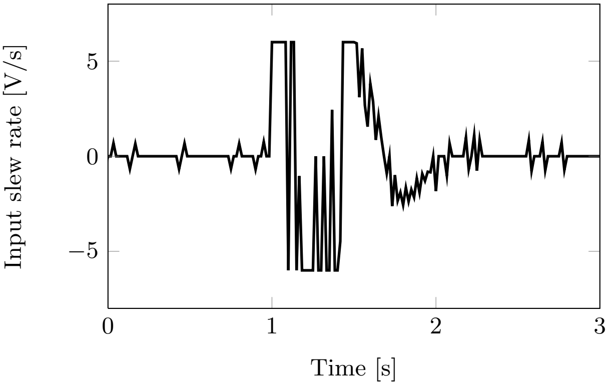

Figure 3.23: Differential input applied to the overhead crane for a requested step of 10 cm controlled by a TOMPC controller.

A second set of experiments illustrates the TOMPC controller disturbance rejection capabilities. Fig. 3.24 shows the position of the trolley x and Fig. 3.25 the swing angle θ when an external disturbance is applied to the lower mass. From time 1.2 s till 1.5 s, as indicated by the grey zone in Fig. 3.25, the lower mass is manually moved out of its equilibrium position $\theta = 0°$ and is released at time 1.5 s. The controller reacts to this disturbance, and steers the system in minimal time back to its desired zero position. Fig. 3.27 illustrates the time-optimality of the reaction to disturbances by showing the input slew rate, which is again hitting continuously the slew rate constraints. The relative slow reaction is due to the low estimator bandwidth and hence it takes a long time before the TOMPC controller knows the disturbance and can react to it. Note also that the imposed disturbance is approximately 20° and therefore exceeds the linear applicability range of state space model (3.17) showing the inherent robustness of this closed loop controller approach.

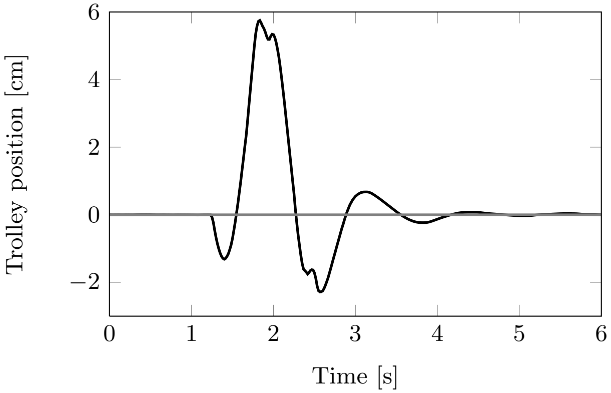

Figure 3.24: Position of the trolley (black line) controlled around the zero position (grey line) by a TOMPC controller when a disturbance is applied.

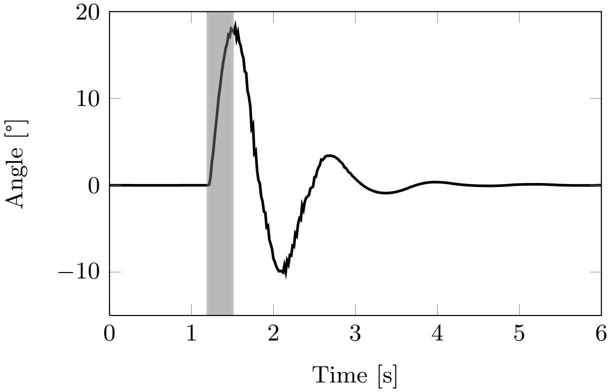

Figure 3.25: Swing angle of the TOMPC controlled system when a disturbance is applied. In the grey zone, the system is manually moved out of its reference position.

Time optimal behavior can also be attained using an optimized reference trajectory (e.g. [Demeulenaere et al., 2009a]) as a feedforward for a linear controller. However with the linear controller approach, time-optimality can only be obtained if for each possible reference step a new reference trajectory optimization is performed,

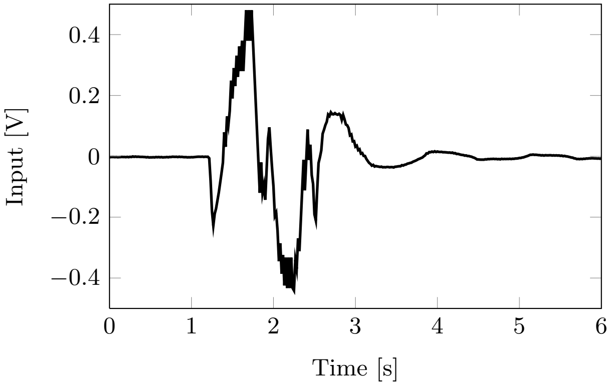

Figure 3.26: Input applied to the overhead crane controlled by a TOMPC controller when a disturbance is applied.

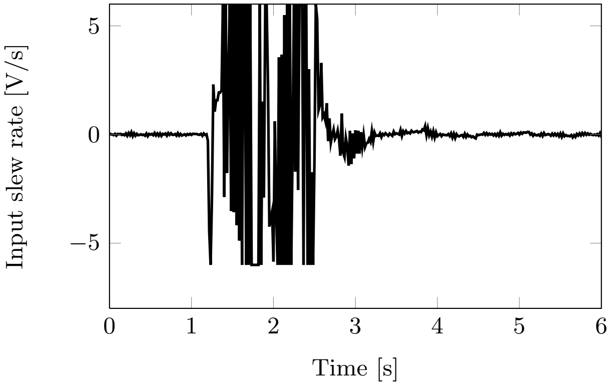

Figure 3.27: Differential input applied to the overhead crane controlled by a TOMPC controller when a disturbance is applied.

whereas the presented TOMPC obtains this behavior for all possible references. This is illustrated in the following figures. Figs. 3.28, 3.29 and 3.30 show respectively the position of the trolley x, the swing angle θ and the input u for several reference steps ranging from 20cm to 50cm. Fig. 3.30 and more clearly Fig. 3.31 show that the controller is continuously hitting the input constraints during motion, showing its time-optimality for all reference steps.

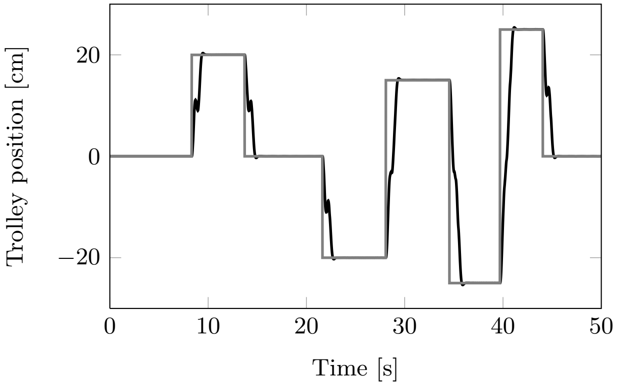

Figure 3.28: Position of the trolley (black line) controlled by a TOMPC controller for a series of reference steps (grey line).

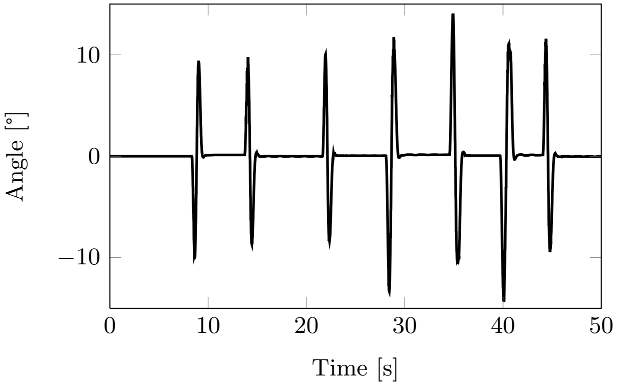

Figure 3.29: Swing angle of the TOMPC controlled system for a series of reference steps.

3.4 Conclusions

This chapter discusses the design and experimental validation of a special type of MPC, called 'time optimal MPC' (TOMPC). A TOMPC controller is a closed loop controller which minimizes the settling time for point-to-point motions while respecting the system constraints. The TOMPC controller extends the idea of the predictive prefilter to include system feedback in order to be able to reject disturbances.

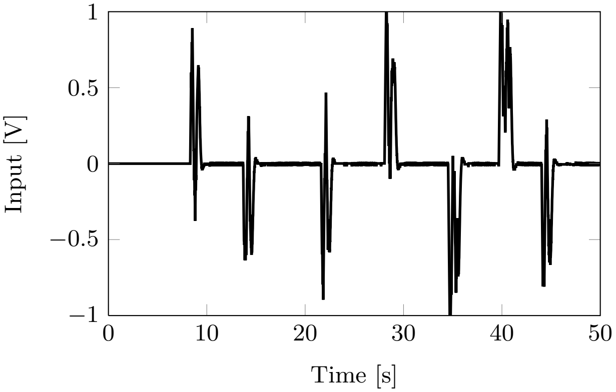

Figure 3.30: Input applied to the overhead crane controlled by a TOMPC controller for a series of reference steps.

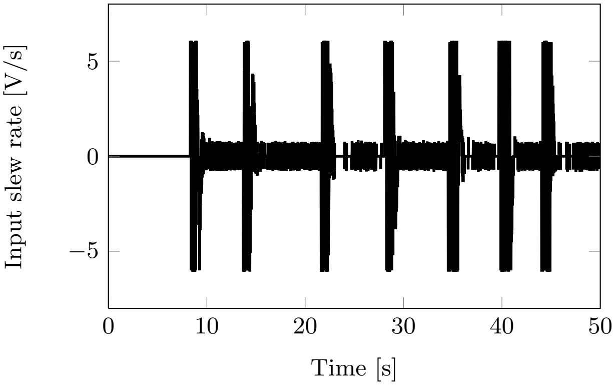

Figure 3.31: Differential input applied to the overhead crane controlled by a TOMPC controller for a series of reference steps.

The main contributions of this chapter are as follows. First, the mixed integer time-optimal optimization problem has been reformulated as a two-level optimization problem which can be solved by a series of feasibility problems. These feasibility problems have been formulated as a traditional MPC problem with a quadratic objective function, system constraints and an endpoint constraint which forces the system to be at rest at the end of the prediction horizon. It is shown that this time-optimal controller is nominally converging towards the desired setpoint. Moreover, by adding an extra parameter to the optimization problem a trade-off between time-optimality and sensitivity to measurement noise is obtained. Through a careful formulation of the optimization problem and implementation of the active set numerical solution method, the maximal solution time has been decreased by a factor of ten. Also, by the use of blocking in the prediction horizon, the attainable range of the controller could be increased by a factor four without compromising the computational speed. As a result, sampling rates up to 200 Hz are achieved for system models with orders up to five and prediction horizons up to forty-five time steps. The TOMPC controller has been validated experimentally on a linear motor drive. For this application, the TOMPC controller is compared both with a linear and an MPC controller. The advantage of the TOMPC controller compared to a normal MPC controller is that true time-optimal behavior is obtained, yielding a reduction of at least 25 % of the settling time on this test setup. In order to obtain time-optimal behavior with a linear controller, the linear controller has to be combined with an optimized feedforward trajectory that is optimized before each point-to-point motion introducing an additional delay. Finally, the disturbance rejection capabilities of TOMPC have been demonstrated experimentally on an overhead crane. An open source C++ implementation of the TOMPC controller has been made publicly available on-line.

Chapter 4

Industrial application: combining TOMPC with high positioning accuracy

This chapter discusses the adaptation for and implementation of TOMPC on an industrial linear motor drive system. First, Section 4.1 presents the industrial setting and the benchmark requirements for the time-optimal control design. Second, Section 4.2 presents an adapted TOMPC control scheme such that the imposed benchmark requirements can be satisfied. This section validates also the designed controller experimentally both for the imposed benchmark requirements and for robustness. Finally, Section 4.3 presents a second time-optimal control scheme. This control scheme alleviates some of the problems observed while implementing the time-optimal control design of Section 4.2.

4.1 Problem statement

The developed time-optimal controller has been validated on and adapted for an industrial test setup during a one-week research stay at ETEL [ETEL S.A., 2010]. ETEL is a Swiss motor manufacturer which produces linear and torque motors. They develop also total motion systems for their clients. For their customers' applications, the cost of a more expensive computer to optimize the system input is often negligible if this allows to increase the throughput and therefore the rendability of e.g. wafersteppers. The setup on which the time-optimal controller is validated at ETEL is a linear motor with air bearings as shown in Fig. 4.1. This is a typical subpart for waferstepper control systems. The input of this system is a reference current i to an internal current loop [A]. The nominal current limitation is 5.5 A and peak currents up to 20 A are attainable. The output of this system is the position of the sliding mass x [m] which is measured by encoders with a resolution of 0.24 nm. This system can be modelled as:

$$\frac{X(z)}{I(z)} = \frac{\alpha}{z^2 - 2z + 1} = \frac{\alpha}{(z-1)^2}. \tag{4.1}$$

The parameter α of this system model is identified based on FRF measurements that are obtained from a full multisine excitation with a frequency content between 1.5 Hz and 25 Hz at a sampling frequency of 2000 Hz [Pintelon and Schoukens, 2001]. Fig. 4.2 shows the measured (black line) and identified FRF (grey line).

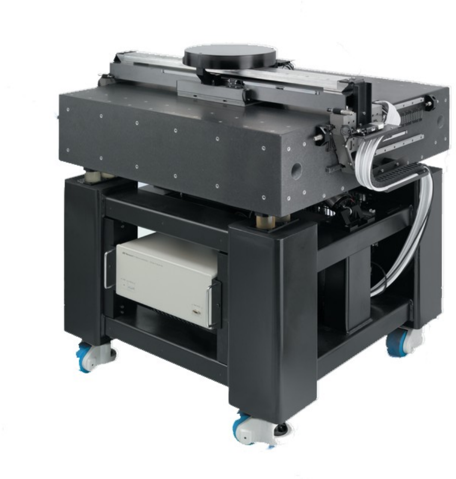

Figure 4.1: Experimental test setup of a linear motor with airbearings.

For this test setup, a benchmark control problem that is typcial for waferstepper applications is defined. This benchmark problem requires the system to make a step of 25 mm with a maximal settling time of 300 ms. For this system, settling time is defined as the time required for arriving at and staying within a band of 100 nm around the desired endposition. Fig. 4.3 illustrates these requirements.

The current ETEL control strategy is to use a PID controller in combination with an S-curve reference trajectory. This control framework is implemented on the ETEL proprietary DSC control system. In order to obtain the desired levels of accuracy, the PID controller needs a sampling frequency of at least a few kilohertz. Due to the agressive nature of the PID controller and in order to avoid actuator saturation,

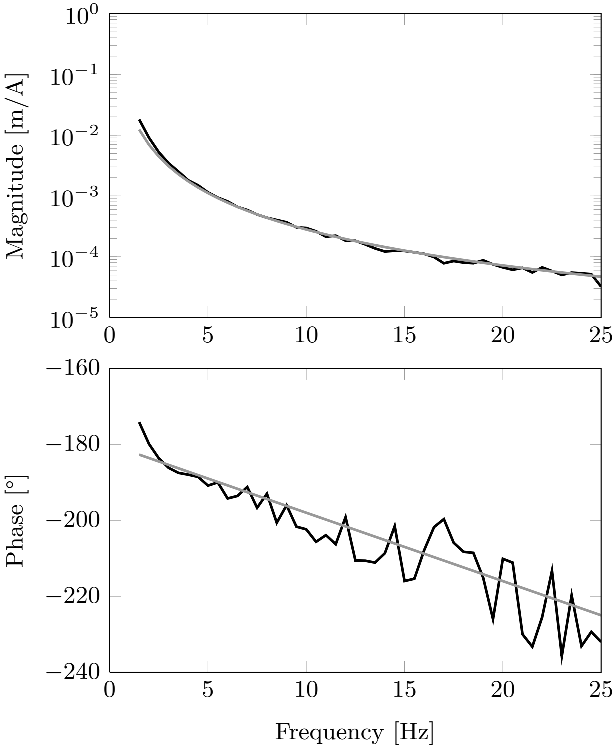

Figure 4.2: Measured FRF (black) and FRF of the identified model (grey) from the current input [A] to the position of the slider [m].

only very small reference steps of a few micrometer can be fed to the PID controller. Therefore, an S-curve smoothens the reference step such that the PID controller does not saturate. The S-curve is computed on-line taking into account constraints on velocity, acceleration and jerk of the reference trajectory. Actuator constraints can however not be taken into account. Application of this control framework results in a controller which satisfies the requirements of the benchmark problem. However, the actuator constraints are not active in this solution. Hence, true time-optimality is not yet obtained and the settling time can still be reduced further. In order to apply more sophisticated controllers, this basic control system can be extended by controllers implemented through C++ functions in Simulink and running on a Speedgoat platform. This setup introduces an extra sample delay. If the ETEL controller is implemented within this control scheme setting, it can only reach an accuracy of 3 µm after 300 ms. Therefore, for a fair comparison a relaxed version of the benchmark settling requirement is defined; i.e. arriving at and staying within a band of 3 µm around the desired setpoint.

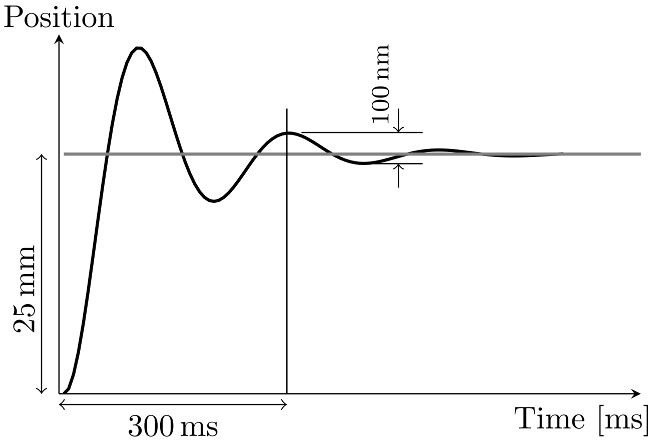

Figure 4.3: Graphical representation of the benchmark requirements on a 25 mm reference step for the linear motor test setup at ETEL.

4.2 TOMPC with high positioning accuracy

This section describes and validates the adapted time-optimal control scheme. First, the new control scheme which allows to combine short settling times with high positioning accuracy, is presented. Then, the implementation and experimental validation of the control scheme is discussed.

4.2.1 New control scheme for the time optimal controller

The benchmark control problem requires not only that the controlled system has a very low settling time. Also a very high positioning accuracy is desired. In order to obtain the specified positioning accuracy, the controller requires a sampling frequency in the order of kilohertz. As such a high sampling frequency is not attainable with the TOMPC controller, a switching control scheme as shown in Fig. 4.4 is proposed. This scheme is analogous to [Mayne and Schroeder, 1997]. In this control scheme, the TOMPC controller is implemented as a feedback controller running at a sampling frequency of 250 Hz. This TOMPC controller is active during most of the movement. However, in a region $\mathbb{X}_0$ close to the setpoint where the high positioning accuracy is required, the TOMPC controller is switched off and a local feedback controller running at a sampling frequency of 12 kHz is switched on. This results in control algorithm 2.

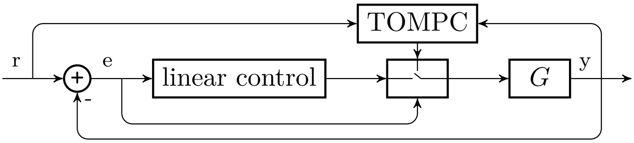

Figure 4.4: Switching control scheme to obtain time-optimal control for the linear motor drive G. The time-optimal controller (TOMPC) runs at 250 Hz and is active during the system movement. Close to the setpoint, the linear controller running at 12 kHz takes over the control action.

Algorithm 2 Optimization procedure for high-accuracy time-optimal control.

```
input: $\bar{x}_l$
parameter: $\mathbb{X}_0$
output: $u^\star$ or error 'infeasible problem'
start with initial guess for $N$
if $x \in \mathbb{X}_0$ then
    apply local linear control
else
    apply time-optimal control algorithm 1
end if
```

4.2.2 Proof of asymptotic convergence and constraint satisfaction

In the linear optimization problem setting, the input and state constraints can typically be expressed as polyhedral constraints $u \in \mathbb{U}$ and $x \in \mathbb{X}$. In order to have a usefull switching time-optimal controller, it is important to have a guaranteed convergence towards the desired settling point and to ensure that the system constraints are respected, i.e. the system input $u \in \mathbb{U}$ and the system state $x \in \mathbb{X}$. The following proof assumes that the system is driven towards the desired setpoint zero. This is no constraint on the applicability of the proof as this zero reference position can always be obtained by a shift in variables.

Assumption 1. *A locally converging controller* $u_k = K_{local} x_k$ *exists.*

Assumption 2. *A local region* $\mathbb{X}_0 \subset \mathbb{X}$ *exists such that if* $x_k \in \mathbb{X}_0$, $u_k = K_{local} x_k \in \mathbb{U}$, *and* $x_{k+1} = Ax_k + Bu_k \in \mathbb{X}_0$.

Theorem 3. *For each* $x_0 = \bar{x}_l \in \mathbb{X}(N_{\max})$, *control algorithm 2 generates a closed-loop response for an undisturbed system without model-plant mismatch, which is asymptotically attracted by* x_{ref}*; i.e.* $\bar{x}_l \to x_{ref}$ *when* $l \to \infty$*. Moreover, the controller always satisfies the system constraints and input constraints.*

Proof.

- If the initial state $x_0 \in \mathbb{X}_0$; The system is controlled by the local controller K_{local}. By assumption 2, this controller is converging to the desired setpoint, and by construction of $\mathbb{X}_0$, the system and input constraints are always satisfied.
- If the initial state $x_0 \in \mathbb{X}/\mathbb{X}_0$; The system is initially controlled by the TOMPC controller described in algorithm 1. This controller respects the system constraints, and under the assumption of no model-plant mismatch is converging towards the desired setpoint as proven in Theorem 2. Therefore, the system state will be driven towards $\mathbb{X}_0$. In this region the local linear controller is converging and respects the system constraints as shown in the first part of proof. □

If no state feedback controller but a dynamic output feedback controller is employed, the proof is still valid if the system state x is extended to also include the internal states of the feedback controller. However, this approach might restrict severely the local region $\mathbb{X}_0$. Another approach which can be employed when dynamic output feedback is used, is to transform this controller to an equivalent state feedback controller. This transformation is possible if the order of the controller is smaller than the order of the system, see eg. [Hartley and Maciejowski, 2009]. Note however that although this proof guarantees constraint satisfaction, the set $\mathbb{X}_0$ can be too small to be attainable by TOMPC in a practical implementation due to disturbances, measurement noise and model-plant mismatch. For these applications either a larger switching region $\mathbb{X}_0'$ can defined which results in constraint violation or the linear controller can be detuned to increase $\mathbb{X}_0$ which results in a lower postioning accuracy.

4.2.3 Experimental validation

Validation

The TOMPC controller is developed with 25 decision variables in order to keep the on-line optimization possible at the sampling frequency of 250 Hz. The prediction horizon is extended by implementing the blocking technique as presented in Section 3.2.3. The internal state estimator is designed by pole placement with poles placed at 100 Hz. To introduce enough insensitivity with respect to measurement noise, the value of $N_{\min}$ is chosen equal to four. The local linear controller provided by ETEL is a PID controller. The corresponding local region $\mathbb{X}_0$ is very small. In practice, it is not guaranteed that the system will reach this local region $\mathbb{X}_0$ due to the relatively low sampling frequency of the TOMPC controller of 250 Hz and model-plant mismatch. Therefore the switching between the TOMPC controller and the PID controller takes place when the system is within a band of output accuracy of 50 µm. This band has been chosen by trial-and-error as being the smallest band which can be guaranteed that to be reached. This control scheme has been implemented in Simulink and runs on the Speedgoat platform. In this implementation, the necessary sampling frequency transformations are included by ‘rate transition blocks’ in order to keep all operations at different sampling frequencies synchronized as shown in Fig. 4.5.

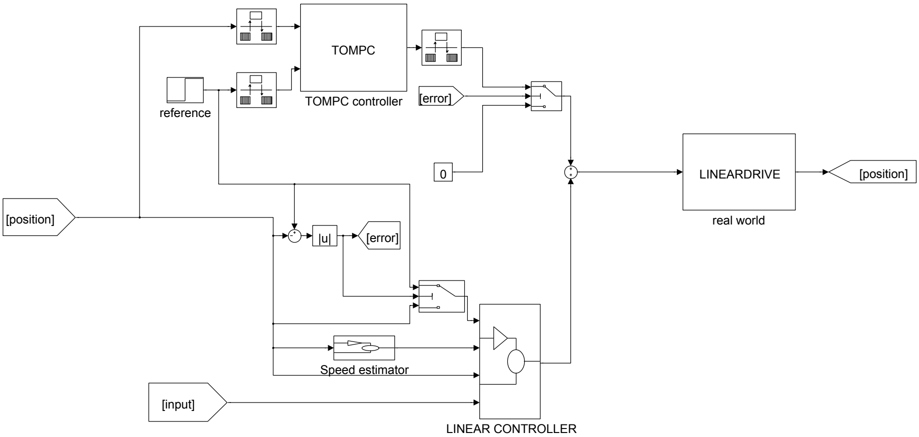

Figure 4.5: Implementation of the TOMPC controller at ETEL in Simulink. Necessary frequency transformations are included to keep all controllers synchronised. Depending on the absolute output error, the system is driven by the TOMPC or PID controller.

A first test verifies whether the adapted control scheme satisfies the benchmark problem requirements. Fig. 4.6 shows the position of the linear motor drive (black solid line) if at time 0.1 s a reference step of 25 mm is requested (grey solid line). Fig. 4.7 shows the corresponding absolute output error (black solid line) on a logarithmic scale. This figure also shows the two benchmark accuracies of respectively 3 µm (grey solid line) and 100 nm (grey dashed line). This figure illustrates clearly that the implemented control strategy satisfies easily the benchmark requirement of a settling time of 300 ms with a settling accuracy of 3 µm. Moreover, also the original requirement of a settling accuracy of 100 nm after 300 ms is satisfied.

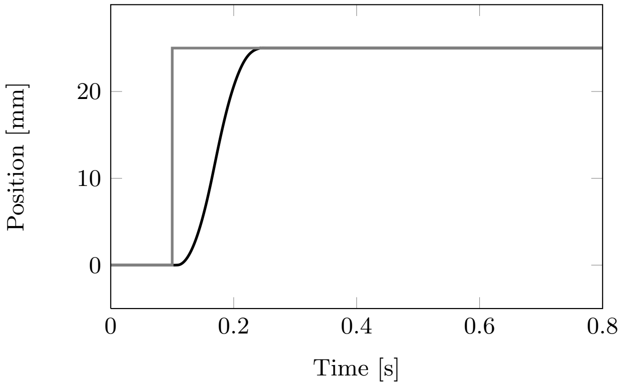

Figure 4.6: Position of the linear motor drive (black solid line) and the reference trajectory (grey solid line) for a reference step of 25 mm at time 0.1 s.

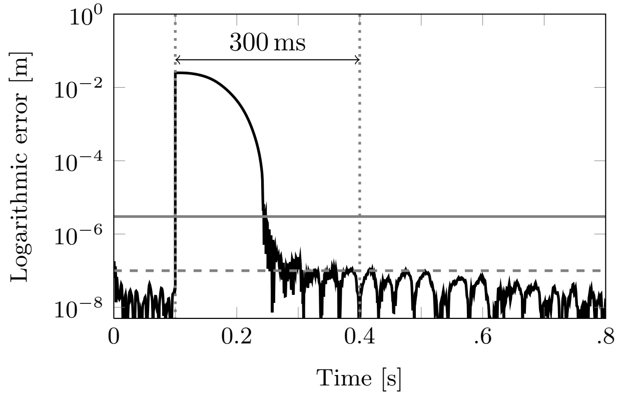

Figure 4.7: Absolute output error of the position of the linear motor on a logarithmic scale (black solid line) for a reference step of 25 mm at time 0.1 s. The benchmark accuracies to be obtained after 0.3 s of movement are respectively 3 µm (grey solid line) and 100 nm (grey dashed line).

If a controller is developed for a serial production, it is important to have an off-the-shelf controller which is stable and preferably also performs well with respect to the design requirements. Therefore, in a second experiment, it has been verified whether the designed controller is robust with respect to model-plant mismatch. For this experiment, three controllers have been developed; one based on the nominal plant dynamics, one based on the dynamics of a plant with an assumed mass which is 10 % heavier than the nominal mass and one based on the dynamics of a plant with an assumed mass which is 10 % lighter than the nominal mass.

These three controllers have been implemented in Simulink and they have been used to control the original system. Fig. 4.8 shows the resulting output behavior for the controller developed with the nominal model (black dashed line), the lighter model (black solid line) and the heavier model (black dotted line). Also, the two benchmark accuracy lines of respectively 3 µm (grey solid line) and 100 nm (grey dashed line) are presented. This shows clearly that the resulting controllers still satisfy the benchmark requirements even for the considered high levels of model-plant mismatch, which would almost never occur for high-end products.

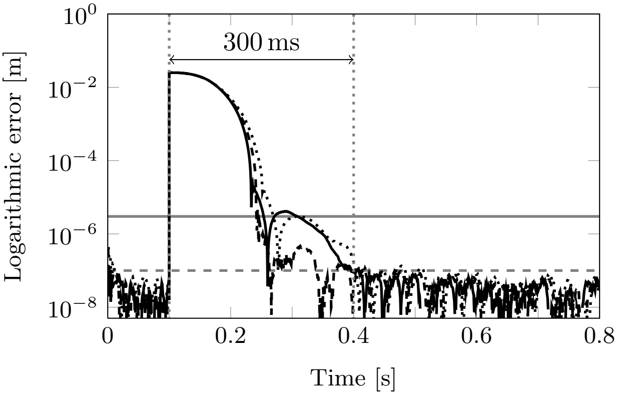

Figure 4.8: Absolute output error of the position of the linear motor on a logarithmic scale for a reference step of 25 mm at time 0.1 s. The errors are given for the nominal controller (black dashed line) and for controllers designed for a system which is respectively 10 % lighter (black solid line) and 10 % heavier (black dotted line) than the nominal system. The benchmark errors are respectively 3 µm (grey solid line) and 100 nm (grey dashed line).

Discussion

In comparison with the current control approach used at ETEL, the advantage of the developed TOMPC based control scheme is that time-optimal behavior is explicitly imposed and not indirectly by a smoothened reference trajectory. Moreover, input constraints can be dealt with directly. Therefore, the system can really be driven towards its nominal constraints and true time-optimality is obtained.

A disadvantage of this adapted control scheme is the switching between the TOMPC controller and the local PID controller, especially as this occurs at an error level where the PID controller still saturates. The resulting system input exceeds the nominal current constraint during a relatively long period and even exceeds the peak current limit of 20 A during a very brief period at 0.24 s as illustrated in Fig.

4.9. This figure shows the control input corresponding to the output shown in Fig. 4.6. This undesirable input behavior is tackled by a second adapted control scheme.

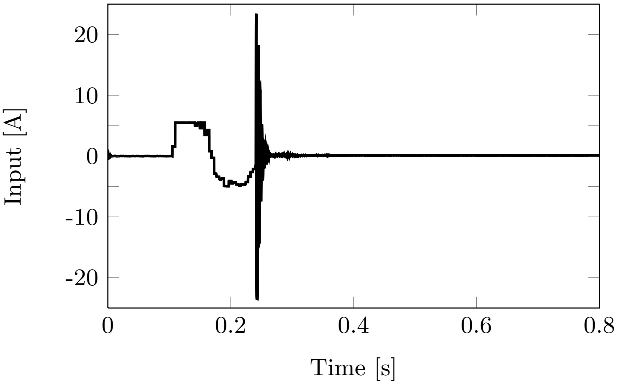

Figure 4.9: Control input current of the linear motor actuators when a step reference of 25 mm is requested at time 0.1 s and the switching between TOMPC and PID control occurs at time 0.24 s.

4.3 TOMPC with high positioning accuracy: scheme 2

If the local set $\mathbb{X}_0$ is too small to be practically relevant, the problems of input saturation of the switching control scheme as presented in Section 4.2 can not be avoided. One approach to circumvent this problem is to design a different local controller with a bigger linear control zone $\mathbb{X}_0$. These linear controllers have typically a lower gain and bandwidth, which for some applications is not desirable. For these applications, a second control scheme is proposed.

As this control scheme still has to satisfy the requirements on settling accuracy and minimization of settling time, both the linear controller and the TOMPC concept are kept in the new scheme. In order to avoid saturation of the linear controller, this controller has to be fed with limited step references. Therefore, a second control scheme as shown in Fig. 4.10 is proposed. In this scheme, the TOMPC controller running at 250 Hz is a pure feedforward controller which generates a feedforward signal u_{ff}. This feedforward signal is obtained by algorithm 1 where the initial system state at every time-sample is provided by system simulation. The PID controller is only active and hence provides feedback action u_{fb} if there is a difference e between the actual output y and the simulated output $\tilde{y}$. This simulated output is obtained by a system simulator $\tilde{G}$. This system simulator runs at the same sampling frequency as the linear controller, i.e. 12 kHz for the

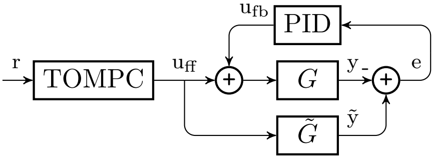

Figure 4.10: Adapted control scheme developed at KULeuven. The time optimal controller runs at 250 Hz. The PID controller and the simulator $\tilde{G}$ run both at 12 kHz.

considered control application, and not at the sampling frequency of the TOMPC controller. As the TOMPC controller runs at a relatively low sampling frequency, using its simulated system output as a reference $\tilde{y}$ for the linear controller, would result in steps on the output error e that are too big and which would saturate this linear controller. Fig. 4.11 illustrates this by showing the total input signal (grey line) and feedforward input (black line) if the simulator $\tilde{G}$ runs at the same frequency as the TOMPC controller.

Note that the desired control objective of control scheme in Fig. 4.10 is to track the reference position r with system G, i.e. a zero off-set steady state error $(r - y)$ is desired. In order to obtain this, it is of the utmost importance that the internal model of the TOMPC controller and the simulator $\tilde{G}$ are consistent, i.e. they have the same DC gain. Also, as the TOMPC controller does not receive sensor feedback and hence does not get noise corrupted state estimates, the value of $N_{\min}$ can and should be reduced.

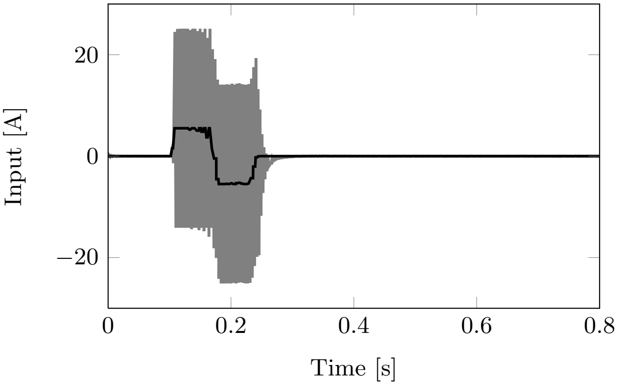

Figure 4.11: Total control input $(u_{\text{ff}} + u_{\text{fb}})$ (grey solid line) and feedforward input (u_{ff}) (black solid line) for a reference step of 25 mm imposed at time 0.1 s, obtained using the control scheme of Fig. 4.10 when the system simulator $\tilde{G}$ runs at 250 Hz.

The adapted control scheme and the control scheme presented in Section 4.2 are validated and compared numerically using a model of the considered linear motor system. The model representing the system and the model used in the TOMPC scheme are the same except for the gain which differs 5 %. Also, an output disturbance of 10 µm and measurement noise of 10 nm have been added to the system.

Fig. 4.12 shows the simulated system output for a reference step of 25 mm (grey solid line) for both control schemes. Both output behaviors are comparable. Fig. 4.13 illustrates this in more detail by showing the logarithmic value of the absolute output error for the two control schemes. This figure illustrates that the new control scheme (black solid line) obtains the same absolute accuracy as the control scheme developed in Section 4.2 (black dashed line) if the simulator and TOMPC model are consistent. Moreover, a comparable settling time is obtained. Fig. 4.14 shows that the total input (grey solid line), i.e. the sum of the feedforward input and feedback input obtained with the feedforward control scheme, does not saturate and does not differ much from the feedforward (black solid line) for these levels of model-plant mismatch, measurement noise and disturbances. This in contrast with the simulated input obtained by using the scheme as presented in Section 4.2 which is given in Fig. 4.15.

A disadvantage of this second control scheme is that the TOMPC does not determine directly the control input. Therefore, contraint satisfaction is not guaranteed. However, this can be mitigated in the following ways. First, if the feedforward control signal is limited to the system's nominal input constraints, these constraints can be temporarily exceeded up to the peak currents. Therefore, if the expected model-plant mismatch and disturbances are limited and this buffer is expected to be sufficient, the feedforward can use all input up to the nominal constraints. Second, if the input constraints are tight, the effects of input saturation can be mitigated by imposing artificially lower constraints during optimization and thereby creating an input buffer zone. Third, if limits on the disturbances are known, these can be incorporated in the optimization procedure. Note however that the last two approaches introduce conservativeness and therefore limit the time-optimal behavior in comparison with a TOMPC controller.

4.4 Conclusion

This chapter discusses the application of time-optimal control on an industrial linear motor setup with benchmark requirements with respect to positioning accuracy and settling time that correspond to the control of waferstepper applications. In order to obtain the required positioning accuracy, a linear controller running at a sampling frequency in the kilohertz range is required. Therefore, two adapted

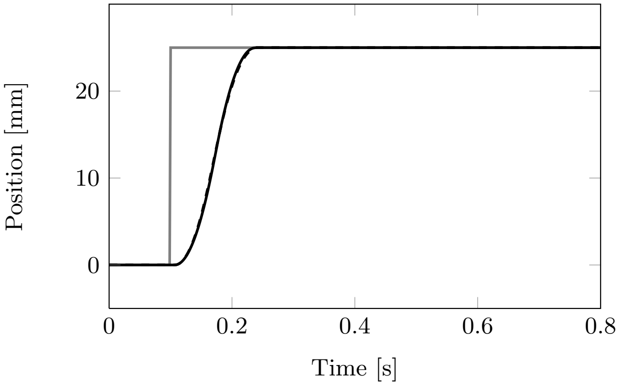

Figure 4.12: Simulated output of the linear motor drive using the control scheme developed at ETEL (black dashed line) and the adapted control scheme (black solid line) and the reference trajectory (grey solid line) for a reference step of 25 mm at time 0.1 s.

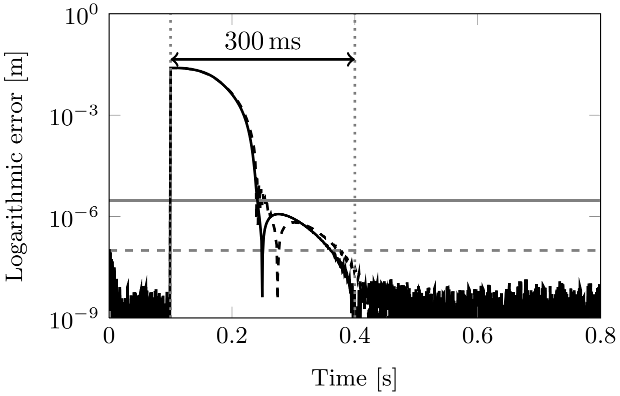

Figure 4.13: Absolute simulated output error of the position of the linear motor on a logarithmic scale both with the control scheme developed at ETEL (black dashed line) and the feedforward control scheme (black solid line) for a reference step of 25 mm at time 0.1 s. The original benchmark errors are shown for completeness and are put at respectively 3 µm (grey solid line) and 100 nm (grey dashed line).

control schemes have been presented in order combine time-optimality with this linear controller.

The main contributions of this chapter are as follows. A first control scheme which switches between a TOMPC controller running at a relative low sampling

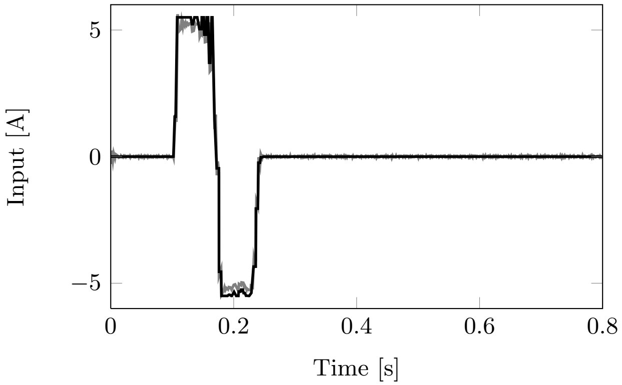

Figure 4.14: Simulated total input (grey solid line) and simulated feedforward input (black solid line) for a reference step of 25 mm requested at time 0.1 s obtained with the feedforward control scheme where the system model $\tilde{G}$ runs at 12 kHz.

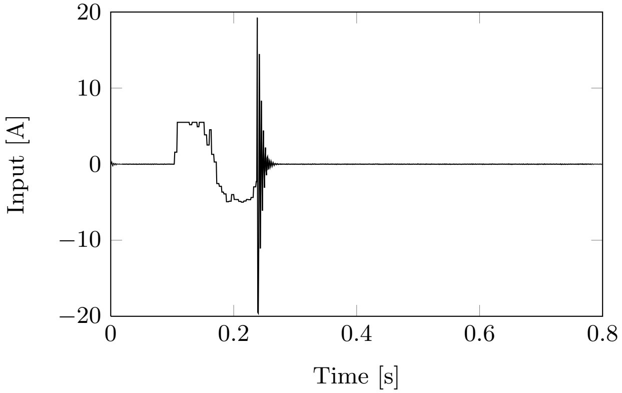

Figure 4.15: Simulated total input for a reference step of 25 mm requested at time 0.1 s obtained with the switching control scheme.

frequency and a linear feedback controller running at a high sampling frequency, has been presented. It has been proven that this control scheme is nominally converging and that this controller always satisfies the system constraints. This controller has been validated experimentally during a one week research stay at ETEL. These experiments show that the developed control scheme satisfies all benchmark requirements. Also, experiments show that this control scheme is robust with respect to model-plant mismatch. However, these experiments also show that the conditions which guarantee constraint satisfaction are too restrictive such that in an actual implementation input violation can occur. Therefore, a second control scheme has been proposed. In this control scheme, the time-optimal controller generates a feedforward signal that is supplied to a system simulator that runs at the same high sampling frequency as the linear controller, generating a reference signal for the feedback scheme. Feedback in this control scheme is provided by a linear controller which acts on the difference between the system output and the simulated output. Although this scheme can not guarantee input constraints satisfaction, it has been shown numerically that the nominal input constraints are better respected.

Chapter 5

Conclusions and future work

5.1 Conclusions

Within this thesis, time-optimal controllers have been developed for mechatronic systems within the MPC framework. The time-optimal MPC controllers realize a minimization of the settling time for any given step reference by solving a constrained optimization problem. This allows to take into account system constraints as e.g. actuator limitations. One of the main challenges in this approach is solving the optimization problem within a few milliseconds such that sampling frequencies in the range of 100 hertz and more are feasible. These controllers are developed for applications which do not perform repetitive tasks but for which minimization of settling time is nevertheless crucial.

In the design of MPC based time-optimal controllers, the following main results have been obtained. First, the time-optimal problem has been defined within the MPC framework. Then, the underlying optimization problem structure has been exploited by incorporating system knowledge to speed-up the solution of the optimization problem. This allows sampling frequencies up to 250 hertz and optimization problems with up to 45 decision variables. Moreover, it has been proven that these controllers are converging towards the desired setpoint. Last, all controllers have been validated experimentally on representative mechatronic test setups like linear motors and an overhead crane. Hence, all hypotheses stated in Section 1.3 have been met.

Within this general framework, three time-optimal MPC controllers have been developed.

First, the predictive prefilter has been developed as a more performant replacement for traditional linear prefilters. In comparison with these prefilters, time-optimality can be obtained for any possible reference step while respecting the system constraints. The real-time implementation is obtained by approximating the time-optimal problem by one linear optimization problem. Also, robustness with respect to model-plant mismatch is introduced in the framework. This controller has been validated on a mass-spring-damper system with a sampling frequency of 100 hertz.

Second, the time-optimal prefilter is extended to include system feedback in order to be able to reject disturbances. This closed-loop controller solves the mixed-integer time-optimal optimization problem by a feasibility search. In this feasibility search, the problem structure and all a priori knowledge on the solution is exploited. The developed controllers have been validated experimentally on a linear motor drive and an overhead crane which have sampling frequencies of 200 hertz and 60 hertz respectively. Also, the C++ implementation of the controller is made publicly available on-line.

Third, the time-optimal feedback controller is adapted for an industrial linear motor. For this application, not only time-optimality is important, but also a very high absolute positioning accuracy. In order to obtain this accuracy, sampling frequencies in the kilohertz range are required. Therefore, a control scheme is proposed which combines the time-optimal controller with a linear feedback controller. This control scheme has been validated experimentally and satisfies the industrial benchmark requirements on both settling time and settling accuracy. Moreover, the controller proves to be robust with respect to model-plant mismatch. However, input constraint satisfaction when the linear controller is employed should be analyzed further.

Hence, in this thesis three time-optimal control approaches have been developed as an alternative to current linear control strategies and normal MPC controllers. An advantage of the developed controllers in comparison with linear control strategies is the on-line computation of the controller which allows constraint satisfaction and time-optimality for all reference steps and this without introducing a delay between a step request and the start of control action. However, this is at a cost of more expensive real-time hardware. In comparison with normal MPC approaches, the advantage is that true time-optimality can be enforced. Moreover, a first step has been made towards the actual implementation of the developed time-optimal controllers on an industrial setup.

5.2 Recommendations for future work

For many industrial mechatronic applications, time-optimality is a critical issue and this thesis is a first step towards time-optimal MPC controllers. However, there are still several restrictions and shortcomings before they can be actually implemented. Therefore, further research should focus on future application of the developed time-optimal control techniques on real-world setups. This research can be divided into two categories: theoretical and practical extensions.

Theoretical extensions

One very important theoretical extension is a further decrease in required computational time. In this thesis, sampling frequencies up to 250 hertz are obtained. However, a further increase by a factor of ten makes the real application of the time-optimal controllers more probable. In order to this, it should be analyzed how the problem structure can be exploited further. Also, other solution techniques for the underlying optimization problem can be analyzed to see how their properties can be exploited to speed up the optimization algorithm. Especially an analysis of uncondensed QP-solvers [Kirches et al., 2011], seems relevant for the considered optimization problems.

In the current development of the time-optimal controllers, knowledge of the system state has been assumed. However, for practical application of these controllers, the state has to be estimated. The bandwidth of the state estimators does have an influence on the total control performance. Therefore, state estimation should also be analyzed more thoroughly and possibly be incorporated into the total time-optimal optimization problem.

Next, the robustness with respect to model-plant mismatch has not yet been analyzed theoretically. It seems essential to analyze systematically what the corresponding degradation in performance is. Also, stability under model-plant mismatch is an unexplored area of research. Moreover, incorporating model uncertainty in the time-optimal feedback control framework seems necessary.

Analogously, the inclusion of strict output constraints without creating infeasible problems has to be analyzed. Output and state constraints are usually introduced in MPC problems as soft constraints to avoid infeasibility of the optimization problems, i.e. these constraints can be violated if this is necessary to keep the problems feasible. However, by construction of the total time-optimal optimization problem, soft constraints can not be imposed to the system as they will be neglected in the feasibility search. Therefore, the combination of strict output constraints and input constraints is nearly impossible in the current framework.

Another area for future theoretical research is to analyze further how the controller can combine the relative slow sampling times necessary to compute the time-optimal controller with a feedback controller running at high sampling frequencies necessary for the high absolute accuracy. Two approaches which combine these requirements are presented in Chapter 4. However, a more thorough analysis of these methods concerning constraint satisfaction is necessary.

Moreover, an extension of the developed techniques to non-linear systems is for the practical application of the control law an important issue. A first step towards this goal might be the extension of TOMPC to mildly non-linear systems which can be represented by LPV models.

Note that although the time-optimal controllers only have been validated on SISO systems, the current control setups are conceptually perfectly transferrable to MIMO systems.

Practical extensions

Second, the application of time-optimal MPC controllers to industrial systems is still an almost completely unexplored field. The developed time-optimal controllers are an interesting approach for systems which perform non-repetitive tasks and require time-optimality for any possible reference step. However, for many of these applications, the developed MPC controllers might have to be tuned further such that they can serve these applications optimally. Moreover, by applying the controller to real applications, industrial requirements can be analyzed and the further development of the controllers can be directed to meet these requirements.

As presented in Chapter 4, a promising application for TOMPC are linear motors which serve in waferstepper applications. For these applications with relatively long movement times in comparison to the processing time, a reduction in settling time can contribute significantly to the total production time and hence increase the total system throughput and rendability. For these applications, a further analysis of high-accuracy time-optimal control is crucial.

For all machining operations where a load has to be transported from point-to-point, time-optimal controllers seem an interesting control technique. This especially in the rapidly changing production environments where series of products become smaller and pure repetitive tasks hence become less ubiquitous.

Another interesting application area for time-optimal control are pick-and-place machines. If these machines are used in production environments where the requested displacement references are not known beforehand, they can decrease the settling time substantially in comparison with linear approaches. However, depending on the transportable load, non-linear or more robust control approaches have to be developed in order to account for the change in system mass and hence system dynamics.

Time-optimal controllers might also be employed in automated warehouses. In these warehouses, products have to be transported from point-to-point and minimization of the total transportation time is required. On a higher level, optimal transportation strategies have been developed for these applications. The lower level controllers that realize the transportation strategies have to the author's knowledge not yet been taken into account.

Likewise, for overhead cranes time-optimal controllers can be interesting. Currently, these systems are often controlled by input shapers in order to reduce residual oscillatory vibrations and settling time. However, to allow hoisting, non-linear adaptations to the control law are crucial.

Appendix A

Implementation of the TOMPC controller

This appendix presents the manual of the TOMPC controller.

TOMPC manual

Manual of the TOMPC package for time optimal control through MPC
Lieboud Van den Broeck May 8, 2011

1. **Introduction**

   This package contains the C++ implementation of a method which minimizes the settling time of (mechatronic) systems on a step reference. This controller is developed within the MPC framework and called time optimal MPC (TOMPC). This package contains a C++ function which implements the TOMPC controller, a matlab function which defines all variables necessary for control and a Simulink scheme which simulates the controller. This package contains also part of the source code of the package qpOASES which implements the underlying active set QP-solver. The design and implementation of the time optimal controller are in more detail described in [1].

2. **Installation**

   - Download the package TOMPC
   - Unpack the package

- In Matlab, browse to the TOMPC path
- Run *make.m*
- Run a simulation test. This simulation example minimizes the settling time of a second order system subject to constraints on the input and slew rate of the input.
    - Run *tompcpreparation.m*
    - Open *tompcsimulate.mdl*
    - Run *tompcsimulate.mdl*

3. **How to use**

First, the control problem has to be set up. All TOMPC variables have to be defined in the matlab m-file *tompcpresentation.m.* Also, the horizon length, sampling time and number of system states have to be defined in the C++ file *tompc.cpp.*
Second, the *tompc.cpp* has to be compiled by running the file *make.m.* If the sampling time, number of system states and horizon length are kept constant, this compilation has to be done only once.
Third, run the file *tompcpreparation.m* to define all variables and constraints in the matlab workspace. Also, the system dynamics matrices are condensed.
Fourth, run the controller in simulink, *tompcsimulate.mdl.*

4. **Implementation on external hardware**

The developed TOPMC controller has been implemented successfully on a dSpace board DS1103 through the real-time target toolbox in Simulink for the optimal control of an overhead crane [2]. An extra C++ compiler for dSPACE has to be added though in order to be able to compile this controller.
The controller has also been implemented on a Speedgoat target machine through the xpc-target toolbox in Simulink for the optimal control of a linear motor [3].
For these implementations, the following links to the external code has to be added:

- In "configuration parameters → real-time workshop → customcode → include directories"; add the path to the headerfiles of qpOASES.
- In "configuration parameters → real-time workshop → customcode → source files"; add all source files of qpOASES

5. **References**

1 Van den Broeck, L., Diehl, M., Swevers, J. *A Model Predictive Control Approach for Time Optimal Point-to-point Motion Control* IFAC Mechatronics (submitted)

2 Van den Broeck, L., Diehl, M., Swevers, J. *Experimental validation of Time Optimal MPC on a flexible motion system.* Proceedings of the 2011 American Control Conference.

3 Van den Broeck, L., Diehl, M., Swevers, J. *Model predictive control for time-optimal point-to-point motion control.* Proceedings of the 18th Ifac World Congress

Bibliography

M. Alamir and A. Murilo. Swing-up and stabilization of a Twin-Pendulum under state and control constraints by a fast NMPC scheme. *Automatica*, 44(5): 1319–1324, May 2008.

T. Alamo, D.R. Ramirez, D. Muñoz de la Peña, and E.F. Camacho. Min-max MPC using a tractable QP problem. *Automatica*, 43(4):693 – 700, 2007.

A. Amthor, J. Werner, A. Lorenz, S. Zschaeck, and C. Ament. Asymmetric motion profile planning for nanopositioning and nanomeasuring machines. *Proceedings Of The Institution Of Mechanical Engineers Part I-Journal Of Systems And Control Engineering*, 224(I1):79–92, 2010.

MOSEK ApS. *The MOSEK optimisation tools manual*, 2008.

R.A. Bartlett and L.T. Biegler. QPSchur: A Dual, Active Set, Schur Complement Method for Large-scale and Structured Convex Quadratic Programming Algorithm. *Optimization and Engineering*, 7:5–32, 2006.

G. Bashein. A simplex algorithm for on-line computation of time optimal controls. *IEEE Transactions on Automatic Control*, 16(5):479 – 482, oct 1971.

M. D. Baumgart and L. Y. Pao. Discrete time-optimal command shaping. *Automatica*, 43:1403–1409, 2007.

VM Beazel and PH Meckl. Command shaping applied to nonlinear systems with configuration-dependent resonance. In *Proceedings Of The 2005 American Control Conference*, pages 539–544, 2005.

R. Bellman. *Dynamic Programming.* Princeton Univ. Press, 1957.

A. Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic control*, 43:415–419, 1998.

A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic control*, 42:340–349, 1997.

A. Bemporad, F. Borrelli, and M. Morari. Model Predictive Control Based on Linear Programming - The Explicit Solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002a.

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

T. Besselmann, J. Lofberg, and M. Morari. Constrained time-optimal control of linear parameter-varying systems. In *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference.*, pages 6923 –6928, Dec 2009.

R.R. Bitmead, M. Gevers, and V. Wertz. *Adaptive optimal control: the thinking man's GPC*. Prentice Hall, Sydney, 1990.

F. Borrelli, A. Bemporad, and M. Morari. A Geometric Algorithm for Multi-Parametric Linear Programming. *Journal of Optimization Theory and Applications*, 118(3):515–540, 2003.

S. Boyd and L. Vandeberghe. *Convex Optimization*. Cambridge University Press, 2004.

A.E. Bryson and Y.-C. Ho. *Applied optimal control: optimization, estimation, and control*. Blaisdell, New York, 1969.

J Buur. *A theoretical approach to mechatronics design*. PhD thesis, Institute for Engineering Design, Technical University of Denmark, Lyngby, 1990.

M. Canale, L. Fagiano, and M. Milanese. Set Membership approximation theory for fast implementation of Model Predictive Control laws. *Automatica*, 45(1): 45–54, Jan 2009.

A Casavola, E Mosca, and M Papini. Control under constraints: An application of the command governor approach to an inverted pendulum. *IEEE Transactions On Control Systems Technology*, 12(1):193–204, Jan 2004.

W. Chatlatanagulchai, V.M. Beazel, and P.H. Meckl. Command shaping applied to a flexible robot with configuration-dependent resonance. In *2006 American Control Conference* , page 6 pp., June 2006.

H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1218, 1998a.

H. Chen and F. Allgöwer. A computationally attractive nonlinear predictive control scheme with guaranteed stability for stable systems. *Journal of Process Control*, 8(5-6):475–485, 1998b.

Dw Clarke, C Mohtadi, and Ps Tuffs. Generalized Predictive Control .1. The Basic Algorithm. *Automatica*, 23(2):137–148, Mar 1987.

T. Coen, W. Saeys, B. Missotten, and J. De Baerdemaeker. Cruise control on a combine harvester using model-based predictive control. *Biosystems Engineering*, 99(1):47 – 55, 2008.

L. Consolini and A. Piazzi. Generalized bang-bang control for feedforward constrained regulation. *Automatica*, 45(10):2234 – 2243, 2009.

D. Corona and B. De Schutter. Comparison of a linear and a hybrid adaptive cruise controller for a Smart. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 4779–84, 2008.

CF Cutforth and LY Pao. Adaptive input shaping for maneuvering flexible structures. *Automatica*, 40(4):685–693, Apr 2004.

C.R. Cutler and B.L. Ramaker. Dynamic Matrix Control- a computer control algorithm. In *Joint Automatic Control Conference*, 1980.

B. Demeulenaere, J. De Caigny, G. Pipeleers, J. De Schutter, and J. Swevers. Optimal splines for rigid motion systems: benchmarking and extensions. *Transactions of the ASME, Journal of Mechanical Design*, 131, 2009a.

B. Demeulenaere, G. Pipeleers, J. De Caigny, J. Swevers, J. De Schutter, and L. Vandenberghe. Optimal splines for rigid motion systems: a convex programming framework. *Journal of Mechanical Design*, 2009b.

D DeRoover and FB Sperling. Point-to-point control of a high accuracy positioning mechanism. In *Proceedings Of The 1997 American Control Conference, Vols 1-6*, pages 1350–1354, 1997.

C. Desoer and J. Wing. A minimal time discrete system. *Automatic Control, IRE Transactions on*, 6(2):111 – 125, may 1961.

ETEL S.A. Etel: Motion technology. `http://www.etel.ch`, January 2010.

H.J. Ferreau. `qpOASES` – An Open-Source Implementation of the Online Active Set Strategy for Fast Model Predictive Control. In *Proceedings of the Workshop on Nonlinear Model Based Control – Software and Applications, Loughborough*, pages 29–30, 2007.

H.J. Ferreau. *qpOASES User's Manual*, 2007–2009. URL `http://homes.esat.kuleuven.be/~optec/software/qpOASES/download/manual-1.1.pdf`.

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(2):293–301, 2007.

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

M. Fliess, J. Levine, P. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: Introductory theory and examples. Technical report, CAS, 1994.

J Fortgang, W Singhose, JD Marquez, and J Perez. Command shaping for micro-mills and CNC controllers. In *Proceedings of the 2005 American Control Conference, Vols 1-7*, pages 4531–4536, 2005.

G.F. Franklin, D.J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall PTR, 2001.

C.E. García, D.M. Prett, and M. Morari. Model Predictive Control: Theory and Practice – a Survey. *Automatica*, 25:335ff, 1989.

EG Gilbert and I Kolmanovsky. Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal Of Robust And Nonlinear Control*, 9(15):1117–1141, Dec 1999.

EG Gilbert and I Kolmanovsky. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. *Automatica*, 38(12): 2063–2073, Dec 2002.

EG Gilbert, I Kolmanovsky, and KT Tan. Discrete-Time Reference Governors And The Nonlinear Control Of Systems With State And Control Constraints. *International Journal Of Robust And Nonlinear Control*, 5(5):487–504, Aug 1995.

P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints. *ACM Transactions on Mathematical Software*, 10(3):282–298, 1984.

P. J. Goulart, E.C. Kerrigan, and J.M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42:523–533, 2006.

P. J. Goulart, E. C. Kerrigan, and D. Ralph. Efficient robust optimization for robust control with constraints. *Mathematical Programming*, 114(1):115 – 147, 2008.

A. Grancharova and T. A. Johansen. Computation, approximation and stability of explicit feedback min-max nonlinear model predictive control. *Automatica*, 45 (5):1134–1143, May 2009.

P. Grieder and M. Morari. Complexity reduction of receding horizon control. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, pages 3179–3190 Vol.3, Dec. 2003.

E.N. Hartley and J.M. Maciejowski. Initial tuning of predictive controllers by reverse engineering. In *Proc. European Control Conference*, Budapest, August 2009.

D Henrion and JB Lasserre. LMIs for constrained polynomial interpolation with application in trajectory planning. *Systems & Control Letters*, 55(6):473–477, Jun 2006.

B. Houska, H.J. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, page (submitted), 2010.

J.R. Huey, K.L. Sorensen, and W.E. Singhose. Useful applications of closed-loop signal shaping controllers. *Control Engineering Practice*, 16(7):836–846, Jul 2008.

J.L. Jerez, G.A. Constantinides, and E.C. Kerrigan. FPGA implementation of an interior point solver for linear model predictive control. In *Proceedings 2010 International Conference on Field-Programmable Technology* , pages 316–19, 2010 2010.

T.A. Johansen. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica*, 40(2):293–300, Feb 2004.

C.N. Jones and M. Morari. Polytopic Approximation of Explicit Model Predictive Controllers. *IEEE Transactions On Automatic Control*, 55(11):2542–2553, Nov 2010.

SD Jones and AG Ulsoy. An approach to control input shaping with application to coordinate measuring machines. *Journal Of Dynamic Systems Measurement And Control-Transactions Of The Asme*, 121(2):242–247, Jun 1999.

T Kailath. *Linear systems.* Prentice Hall, 1980.

R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.

S. Keerthi and E. Gilbert. Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints. *IEEE Transactions on Automatic Control*, 32(5):432 – 435, may 1987.

S.S. Keerthi and E.G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293, 1988.

M Kenison and W Singhose. Concurrent design of input shaping and proportional plus derivative feedback control. *Journal Of Dynamic Systems Measurement And Control-Transactions Of The ASME*, 124(3):398–405, Sep 2002.

Mh Kim and S Engell. Speed-Up Of Linear-Programming For Time-Optimal Control. In *Proceedings Of The 1994 American Control Conference, Vols 1-3*, pages 2667–2670, 1994.

SK Kim and DM Tilbury. Trajectory generation for a class of nonlinear systems with input and state constraints. In *Proceedings Of The 2001 American Control Conference*, pages 4908–4913, 2001.

C. Kirches, H. G. Bock, J. P. Schloeder, and S. Sager. Block-structured quadratic programming for the direct multiple shooting method for optimal control. *Optimization Methods & Software*, 26(2):239–257, 2011.

Christian Kirches, Sebastian Sager, Hans Georg Bock, and Johannes P. Schloeder. Time-optimal control of automobile test drives with gear shifts. *Optimal Control Applications & Methods*, 31(2):137–153, MAR-APR 2010.

D Klatte. Lower Semicontinuity Of Optimal Sets In Convex Parametric Optimization. *Mathematical Programming Study*, 10(Feb):104–109, 1979.

O Kostyukova and E Kostina. Analysis of properties of the solutions to parametric time-optimal problems. *Computational Optimization And Applications*, 26(3): 285–326, DEC 2003.

O. I. Kostyukova and E. A. Kostina. Properties of solutions of parametric time-optimal problems in a neighborhood of an abnormal point. *Differential Equations*, 45(6):889–901, Jun 2009.

M. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, Nov 1996.

H. Kwakernaak and J. Smit. Minimum vibration cam profiles. *Journal of Mechanical Engineering Science*, 10(3):219–227, 1968.

Kj Kyriakopoulos and Gn Saridis. Minimum Jerk For Trajectory Planning And Control. *Robotica*, 12(Part 2):109–113, Mar-Apr 1994.

C. La-orpacharapan and L.Y. Pao. Shaped time-optimal feedback control for disk-drive systems with back-electromotive force. *Magnetics, IEEE Transactions on*, 40(1):85 – 96, Jan 2004.

C. La-orpacharapan and L.Y. Pao. Fast and robust control of systems with multiple flexible modes. *Mechatronics, IEEE/ASME Transactions on*, 10(5):521 – 534, Oct 2005.

P Lambrechts, M Boerlage, and M Steinbuch. Trajectory planning and feedforward design for electromechanical motion systems. *Control Engineering Practice*, 13 (2):145–157, Feb 2005.

W. Langson, S.V. Rakovic I. Chryssochoos, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40:125–133, 2004.

J. H. Lee and Z. Yu. Worst-case Formulations of Model Predictive Control for Systems with Bounded Parameters. *Automatica*, 33(5):763–781, 1997.

C Lewin. Motion Control Gets Gradually Better. *Machine Design*, 66(21):90, Nov 1994.

H. Li, M. D. Le, Z. M. Gong, and W. Lin. Motion Profile Design to Reduce Residual Vibration of High-Speed Positioning Stages. *IEEE-ASME Transactions On Mechatronics*, 14(2):264–269, Apr 2009.

D Limon, T Alamo, and EF Camacho. Stable constrained MPC without terminal constraint. In *Proceedings Of The 2003 American Control Conference*, pages 4893–4898, 2003.

FJ Lin, KK Shyu, and CH Lin. Incremental motion control of linear synchronous motor. *IEEE Transactions On Aerospace And Electronic Systems*, 38(3):1011–1022, Jul 2002.

D.G. Luenberg. Introduction To Observers. *IEEE Transactions On Automatic Control*.

S Macfarlane and EA Croft. Jerk-bounded manipulator trajectory planning: Design for real-time applications. *IEEE Transactions On Robotics And Automation*, 19 (1):42–52, Feb 2003.

J.M. Maciejowski. *Predictive control with constraints.* Prentice Hall, 2000.

U. Maeder, F. Borrelli, and M. Morari. Linear offset-free Model Predictive Control. *Automatica*, 45(10):2214–2222, Oct 2009.

S. Mariethoz and M. Morari. Explicit Model-Predictive Control of a PWM Inverter With an LCL Filter. *IEEE Transactions On Industrial Electronics*, 56(2):389–399, Feb 2009.

J.J. Marquez, J. Fortgang, W. Singhose, and J.M. Perez. Optimization of the micromilling process using vibration reduction through command shaping techniques. *Informacion Tecnologica*, 17(6):3–6, 2006.

DL Marruedo, T Alamo, and EF Camacho. Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC. In *Proceedings Of The 2002 American Control Conference*, pages 364–369, 2002.

J. Mattingley and S. Boyd. *Automatic Code Generation for Real-Time Convex Optimization.* Convex Optimization in Signal Processing and Communications, Y. Eldar and D. Palomar, Eds., Cambridge University Press, 2009.

H Maurer, C Buskens, JHR Kim, and CY Kaya. Optimization methods for the verification of second order sufficient conditions for bang-bang controls. *Optimal Control Applications & Methods*, 26(3):129–156, MAY-JUN 2005.

D. Q. Mayne and W. R. Schroeder. Robust time-optimal control of constrained linear systems. *Automatica*, 33(12):2103 – 2118, 1997.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Schockaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

PH Meckl, PB Arestides, and MC Woods. Optimized S-curve motion profiles for minimum residual vibration. In *Proceedings Of The 1998 American Control Conference, Vols 1-6*, pages 2627–2631, 1998.

RH Miller, I Kolmanovsky, EG Gilbert, and PD Washabaugh. Control of constrained nonlinear systems: A case study. *IEEE Control Systems Magazine*, 20(1):23–32, Feb 2000.

M. Morari and J.H. Lee. Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23:667–682, 1999.

M. Morari and E. Zafiriou. *Robust Process Control.* Prentice Hall, Englewood Cliffs, 1989.

Br Murphy and I Watanabe. Digital Shaping Filters For Reducing Machine Vibration. *IEEE Transactions On Robotics And Automation*, 8(2):285–289, Apr 1992.

K.R. Muske and T.A. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12:617–632, 2002.

G. J. L. Naus, J. Ploeg, M. J. G. Van de Molengraft, W. P. M. H. Heemels, and M. Steinbuch. Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach. *Control Engineering Practice*, 18(8):882–892, AUG 2010.

K.D. Nguyen, T.-C. Ng, and I.-M. Chen. On Algorithms for Planning S-curve Motion Profiles. *International Journal Of Advanced Robotic Systems*, 5(1):99–106, Mar 2008.

J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering. Springer, 2006.

T. Ohtsuka. A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control. *Automatica*, 40(4):563–574, 2004.

A. Olabi, R. Bearee, O. Gibaru, and M. Damak. Feedrate planning for machining with industrial six-axis robots. *Control Engineering Practice*, 18(5):471–482, May 2010.

P. Ortner and L. del Re. Predictive control of a diesel engine air path. *IEEE Transactions On Control Systems Technology*, 15(3):449–456, May 2007.

G. Pannocchia. Robust model predictive control with guaranteed setpoint tracking. *Journal of Process Control*, 14:927–937, 2004.

G. Pannocchia and J.B. Rawlings. Disturbance Models for Offset-Free Model-Predictive Control. *AIChE Journal*, 49:426–437, 2003.

L.Y. Pao and G.F. Franklin. Proximate Time-Optimal Control Of 3Rd-Order Servomechanisms. *IEEE Transactions On Automatic Control*, 38(4):560–580, Apr 1993.

LY Pao and MA Lau. Robust input shaper control design for parameter variations in flexible structures. *Journal Of Dynamic Systems Measurement And Control-Transactions Of The Asme*, 122(1):63–70, Mar 2000.

J Park and PH Chang. Learning Input Shaping Technique for non-LTI systems. *Journal Of Dynamic Systems Measurement And Control-Transactions Of The Asme*, 123(2):288–293, Jun 2001.

R. Pintelon and J. Schoukens. *System Identification: A Frequency Domain Approach.* IEEE Press, 2001.

E. N. Pistikopoulos, V. Dua, N. A. Bozinis, A. Bemporad, and M. Morari. On-line optimization via off-line parametric optimization tools. *Computers and Chemical Engineering*, 24:183–188, 2000.

L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishechenko. The mathematical theory of optimal processes. *Interscience*, 4, 1962.

S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. In J.C. Kantor, C.E. Garcia, and B. Carnahan, editors, *Fifth International Conference on Chemical Process Control – CPC V*, pages 232–256. American Institute of Chemical Engineers, 1996.

S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

DR Ramirez, T Alamo, EF Camacho, and DM de la Pena. Min-Max MPC based on a computationally efficient upper bound of the worst case cost. *Journal Of Process Control*, 16(5):511–519, JUN 2006.

J.B. Rawlings. Tutorial Overview of Model Predictive Control. *IEEE Control Systems Magazine*, 20(3):38–52, 2000.

J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design.* Nob Hill, 2009.

JB Rawlings and KR Muske. The Stability Of Constrained Receding Horizon Control. *IEEE Transactions On Automatic Control*, 38(10):1512–1516, Oct 1993.

K.-H. Rew, C.-W. Ha, and Y.-S. Kim. A practically efficient method for motion control based on asymmetric velocity profile. *International Journal Of Machine Tools & Manufacture*, 49(7-8):678–682, Jun 2009.

J Richalet, A Rault, Jl Testud, and J Papon. Model Predictive Heuristic Control - Applications To Industrial Processes. *Automatica*, 14(5):413–428, 1978.

M. J. Robertson and R. S. Erwin. Command shapers for systems with actuator saturation. In *Proceedings of the American Control Conference*, pages 760–765, New York City, Jul 2007.

C Scherer, P Gahinet, and M Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions On Automatic Control*, 42(7):896–911, Jul 1997.

M. Schlemmer and S.K. Agrawal. A computational approach for time-optimal planning of high-rise elevators. *IEEE Transactions on Control Systems Technology*, 10(1):105 –111, jan 2002.

P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43: 1136–1142, 1998.

P.O.M. Scokaert and J.B. Rawlings. Constrained Linear Quadratic Regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1169, 1998.

M. Scott. Time/fuel optimal control of constrained linear discrete systems. *Automatica*, 22(6):711 – 715, 1986.

H. Seguchi and T. Ohtsuka. Nonlinear Receding Horizon Control of an Underactuated Hovercraft. *International Journal of Robust and Nonlinear Control*, 13(3–4):381–398, 2003.

A. Shahzad, E.C. Kerrigan, and G.A. Constantinides. A fast well-conditioned interior point method for predictive control. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 508 –513, dec 2010.

N. C. Singer and W. P. Seering. Preshaping command inputs to reduce system vibration. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 112:76–82, 1990.

N.C. Singer. *Residual Vibration Reduction in Computer Controlled Machines.* PhD thesis, Dept. of Mech. Eng., M.I.T., 1989.

T. Singh. *Optimal reference shaping for dynamical systems: theory and applications.* CRC Press, Taylor and Francis Group, 2010.

W Singhose, E Crain, and W Seering. Convolved and simultaneous two-mode-input shapers. *Iee Proceedings-Control Theory And Applications*, 144(6):515–520, Nov 1997.

W.E. Singhose, W. P. Seering, and N.C. Singer. Residual vibration reduction using vector diagrams to generate shaped inputs. *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, pages 654–659, 1994.

W.E. Singhose, W. P. Seering, and N.C. Singer. Time-optimal negative input shapers. *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, 119:198–205, 1997.

W.E. Singhose, W.P. Seering, and N.C. Singer. Input shaping for vibration reduction with specified insensitivity to modeling errors. In *Proceedings of the Japan-USA Symposium on Flexible Automation - 1996*, pages 307–13 vol.1, 1996.

O.J.M. Smith. Posicast control of damped oscillatory systems. In *Proceedings of the IRE*, pages 1249–1255, 1957.

K. Sorensen and W. Singhose. Oscillatory effects of common hard nonlinearities on systems using two-impulse zv input shaping. In *Proceedings of the 2007 American Control Conference*, pages 5539 –5544, Jul 2007.

K.L. Sorensen, W. Singhose, and S. Dickerson. A controller enabling precise positioning and sway reduction in bridge and gantry cranes. *Control Engineering Practice*, 15(7):825–837, Jul 2007.

K.L. Sorensen, A. Daftari, W.E. Singhose, and K. Hekman. Negative Input Shaping: Eliminating Overcurrenting and Maximizing the Command Space. *Journal Of Dynamic Systems Measurement And Control-Transactions Of The Asme*, 130(6), Nov 2008.

Speedgoat. Speedgoat: Simulation and real-time systems. `http://www.speedgoat.ch/`, January 2010.

L Sungyung, H. D. Stevens, and J. P. How. Input shaping design for multi-input flexible systems. *Journal of dynamic systems, measurement, and control*, 121: 443–447, 1999.

M. Susanu and D. Dumur. Hierarchical predictive control within an open architecture virtual machine tool. *Cirp Annals-Manufacturing Technology*, 55(1): 389–392, 2006.

T.-W. Yoon and D.W. Clarke. Receding-horizon predictive control with exponential weighting. *International Journal of Systems Science*, 24(9):1745–57, Sep 1993.

M. Tomizuka. Mechatronics: from the 20th to 21st century. *Control Engineering Practice*, 10(8):877 – 886, 2002. ISSN 0967-0661.

P Tondel, TA Johansen, and A Bemporad. Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950, May 2003.

DM Tsay and CF Lin. Asymmetrical inputs for minimizing residual response. In *2005 IEEE International Conference on Mechatronics*, pages 235–240, 2005.

TD Tuttle and WP Seering. Experimental verification of vibration reduction in flexible spacecraft using input shaping. *Journal Of Guidance Control And Dynamics*, 20(4):658–664, Jul-Aug 1997.

L. Van den Broeck, J. De Caigny, G. Pipeleers, and al. A linear programming approach to robust input shaping. In *Proceedings of the AMC*, pages 80–85, Trento, 25-27 March 2008.

L. Van den Broeck, M. Diehl, and J. Swevers. Embedded optimization for input shaping. *IEEE Transactions on Control System Technology*, 18(5):1146–1154, 2010.

L. Van den Broeck, M. Diehl, and J. Swevers. A model predictive control approach for time optimal point-to-point motion control. *IFAC Mechatronics*, 2011. submitted.

J. Vaughan and W. Singhose. Input Shapers for Reducing Overshoot in Human-Operated Flexible Systems. In *Proceedings Of The 2009 American Control Conference*, pages 178–183, 2009.

J. Vaughan, A. Yano, and W. Singhose. Comparison of robust input shapers. *Journal Of Sound And Vibration*, 315(4-5):797–815, Sep 2008a.

J. Vaughan, A. Yano, and W. Singhose. Performance comparison of robust negative input shapers. In *2008 American Control Conference, Vols 1-12*, pages 3257–3262, 2008b.

J. Vaughan, E. Maleki, and W. Singhose. Advantages of using command shaping over feedback for crane control. In *Proceedings of the 2010 American Control Conference*, pages 2308–13, 2010.

D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl. Time-Optimal Path Tracking for Robots: a Convex Optimization Approach. *IEEE Transactions on Automatic Control*, 54:2318–2327, 2009a.

D. Verscheure, M. Diehl, J. De Schutter, and J. Swevers. Recursive log-barrier method for on-line time-optimal robot path tracking. In *Proceedings of the 2009 American Control Conference*, pages 4134–4140, St. Louis, USA, 2009b.

LP Wang. Use of exponential data weighting in model predictive control design. In *Proceedings Of The 40Th IEEE Conference On Decision And Control, Vols 1-5*, pages 4857–4862, 2001.

Y. Wang and S. Boyd. Fast Model Predictive Control Using Online Optimization. *IEEE Transactions On Control Systems Technology*, 18(2):267–278, Mar 2010.

A. Wills, G. Knagge, and B. Ninness. Fast linear model predictive control via custom integrated circuit architecture. *IEEE Transactions on Control Systems Technology*, pages 1–12, aug 2010.

A.G. Wills, D. Bates, A.J. Fleming, B. Ninness, and S.O.R. Moheimani. Application of MPC to an active structure using sampling rates up to 25khz. In *Proceedings of the Conference on Decision and Control and European Control Conference ECC'05*, Seville, 2005.

A.G. Wills, D. Bates, A.J. Fleming, B. Ninness, and S.O.R. Moheimani. Model predictive control applied to constraint handling in active noise and vibration control. *Control Systems Technology, IEEE Transactions on*, 16(1):3 –12, Jan 2008.

M. L. Workman and G. F. Franklin. Implementation of adaptive proximate time-optimal controllers. In *Proceedings of the 1988 American Control Conference*, pages 1629 –1635, Jun 1988.

M. L. Workman, R. L. Kosut, and G. F. Franklin. Adaptive proximate time-optimal servomechanisms: Discrete-time case. In *Proceedings of the 26th IEEE Conference on Decision and Control, 1987.*, volume 26, pages 1548 –1553, Dec 1987a.

M. L. Workman, R. L. Kosut, and G. F. Franklin. Adaptive proximate time-optimal servomechanisms: Continuous time case. In *Proceedings of the 1987 American Control Conference*, pages 589 –594, Jun 1987b.

V. Yousefzadeh, A. Babazadeh, B. Ramachandran, E. Alarcon, L.Y. Pao, and D. Maksimovic. Proximate time-optimal digital control for synchronous buck DC-DC converters. *IEEE Transactions On Power Electronics*, 23(4):2018–2026, Jul 2008.

La Zadeh. On Optimal Control And Linear Programming. *IRE Transactions On Automatic Control*, AC 7(4):45–46, 1962.

E. Zafiriou. Robust model predictive Control of processes with hard constraints. *Computers & Chemical Engineering*, 14(4–5):359–371, 1990.

R Zanasi and R Morselli. Discrete minimum time tracking problem for a chain of three integrators with bounded input. *Automatica*, 39(9):1643–1649, Sep 2003.

R Zanasi, CG Lo Bianco, and A Tonielli. Nonlinear filters for the generation of smooth trajectories. *Automatica*, 36(3):439–448, Mar 2000.

M.N. Zeilinger, C.N. Jones, and M. Morari. Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Optimization. *IEEE Transactions on Automatic Control*, jul 2011.

J. Zhao, M. Diehl, R. Longman, H.G. Bock, and J.P. Schlöder. Nonlinear Model Predictive Control of Robots Using Real-Time Optimization. In *Proceedings of the AIAA/AAS Astrodynamics Conference*, Providence, RI, August 2004.

C. Zheng, Y. Su, and P.C. Mueller. Simple online smooth trajectory generations for industrial systems. *Mechatronics*, 19(4):571–576, Jun 2009.

J. Zhou, R. Zhou, Y. Wang, and G. Guo. Improved proximate time-optimal sliding-mode control of hard disk drives. *Control Theory and Applications, IEE Proceedings -*, 148(6):516 –522, nov 2001.

T. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, 1995.

J.G. Ziegler and N.B. Nichols. Optimum settings for automatic controllers. *Transactions of the ASME*, 1942.

Curriculum Vitae

Personal data

Lieboud Van den Broeck
born: July 5, 1984 in Hasselt (Belgium)
lieboud@gmail.com

Education

2007-2011: PhD student at the Department of Mechanical Engineering, Katholieke Universiteit Leuven (Belgium) as a fellow of the Research Foundation- Flanders (FWO-Vlaanderen). Supervised by Prof. Jan Swevers and Prof. Moritz Diehl.

2005-2007: Master in Mechanical Engineering (summa cum laude), specialization Mechatronics, Katholieke Universiteit Leuven (Belgium). Master thesis: "Optimal prefilters for highly dynamic motion trajectories: development of a generic and numerically efficient design methodology". Supervised by Prof. Jan Swevers and Prof. Joris De Schutter.

April-August 2006: Student exchange to Keio University (Japan) in the DeMaMech exchange program.

2002-2005: Bachelor in Engineering (summa cum laude with congratulations of the board), Katholieke Universiteit Leuven (Belgium).

1996-2002 Secondary School (Latijn-Wiskunde) at the Virga Jesse college, Hasselt

List of publications

Articles in international journals

1. Van den Broeck, L., Diehl, M., Swevers, J. (submitted). A Model Predictive Control Approach for Time Optimal Point-to-point Motion Control. In: *IFAC Mechatronics*
2. Van den Broeck, L., Diehl, M., Swevers, J. (2010). Embedded Optimization for Input Shaping. In: *IEEE Transactions on Control Systems Technology*, 18(5), 1146-1154.

Articles in international conference proceedings

1. Van den Broeck, L., Diehl, M., Swevers, J. (2011). Model predictive control for time-optimal point-to-point motion control. In: *Proceedings of the 18th IFAC World Congress.* Milan, August 28 - September 2, 2011.
2. Van den Broeck, L., Diehl, M., Swevers, J. (2011). Experimental validation of Time Optimal MPC on a flexible motion system. In: *Proceedings of the 2011 American Control Conference.* San Francisco, June 29 - July 1, 2011.
3. Van den Broeck, L., Diehl, M., Swevers, J. (2010). Experimental validation of Time Optimal MPC on a linear drive system. In: *Proceedings of the 11th International Workshop on Advanced Motion Control.* Nagaoka, March 21-24, 2010 (pp. 355-360).
4. Depraetere, B., Pinte, G., Van den Broeck, L., Swevers, J. (2010). A two-level optimization based learning control strategy for wet clutches. In: *Proceedings of the IFAC conference on Adaptation and Learning in Control and Signal Processing (ALCOSP 2010).* Antalya, August 26-28, 2010.
5. Van den Broeck, L., Diehl, M., Swevers, J. (2009). Time Optimal MPC for mechatronic applications. In: *Proceedings of the 48th IEEE Conference on Decision and Control.* Shanghai, December 16-18, 2009 (pp. 8040-8045).

6. Van den Broeck, L., Diehl, M., Swevers, J. (2009). Performant design of an input shaping prefilter via embedded optimization. In: *Proceedings of the 2009 American Control Conference.* St.Louis, June 10-12, 2009 (pp. 166-171).
7. Van den Broeck, L., Pipeleers, G., De Caigny, J., Demeulenaere, B., Swevers, J., De Schutter, J. (2008). Inputshaping: a linear programming approach. In: *Proceedings of the International Conference on Noise and Vibration Engineering.* Leuven, September 15-17, 2008 (pp. 397-408).
8. Van den Broeck, L., Pipeleers, G., De Caigny, J., Demeulenaere, B., Swevers, J., De Schutter, J. (2008). A linear programming approach to design robust input shaping. In: *Proceedings of the 10th International Workshop on Advanced Motion Control.* Trento, March 26-28, 2008 (pp. 80-85).

Abstracts in international conference proceedings

1. Van den Broeck, L., Diehl, M., Swevers, J. (2011). Application of Time Optimal MPC on an industrial linear motor. In: *Proceedings of the 30th Benelux Meeting on Systems and Control.* Lommel, March 15-17, 2011 (pp. 29).
2. Van den Broeck, L., Diehl, M., Swevers, J. (2010). Time optimal MPC for mechatronic systems. In: *Proceedings of the 29th Benelux Meeting on Systems and Control.* Heeze, March 30-April 1, 2010 (pp. 121).
3. Van den Broeck, L., Swevers, J., Diehl, M. (2009). Time Optimal MPC. In: *Proceedings of the Belgian-French-German Conference on Optimization.* Leuven, September 14-18, 2009. (pp. 142).
4. Van den Broeck, L., Diehl, M., Swevers, J. (2009). Predictive input shaping prefilters. In: *Proceedings of the 28th Benelux Meeting on Systems and Control.* Spa, March 16-18, 2009. (pp. 146).
5. Van den Broeck, L., Pipeleers, G., De Caigny, J., Demeulenaere, B., Swevers, J., De Schutter, J. (2008). A linear programming reformulation of input shaping. In: *Proceedings of the 27th Benelux Meeting on Systems and Control.* Heeze, March 18-20, 2008 (pp. 59).