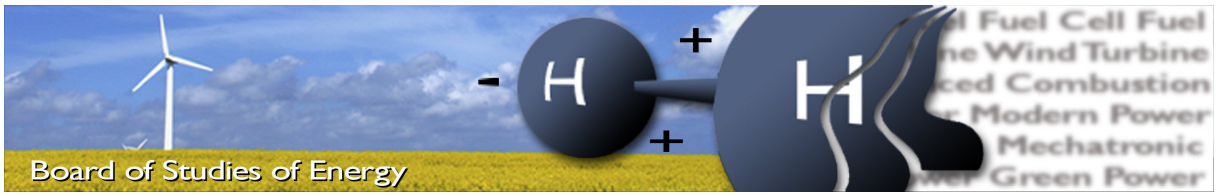
Secondary Control of Discrete Displacement Cylinder

Anders Nielsen
Jens-Kristian Langkjær
Carl Richter

AALBORG UNIVERSITY
STUDENT REPORT

Master of Science (M.Sc.) in Engineering
Mechatronic Control Engineering
School of Engineering and Science
Aalborg University
June 1st 2016



Title:	Secondary Control of Discrete Displacement Cylinder
Semester:	MCE10
Semester theme:	Master's Thesis
Project period:	01.02.16 to 01.06.16
ECTS:	30
Supervisors:	Torben Ole Andersen, Prof., M.Sc., Ph.D.
	Anders Hedegaard Hansen, Ass. Prof., M.Sc., Ph.D.
Project group:	MCE4-1021

Anders Nielsen

Jens-Kristian Langkjær

Carl Richter

SYNOPSIS:

This thesis concerns the investigation of secondary control for a discrete displacement cylinder. The hypothesis is that it is possible to develop optimal control, which balances tracking error and energy losses associated to switching of cylinder chamber pressure level. Existence of limit-cycles is shown, caused by the discrete nature of the system. Non-linear Model Predictive Control is developed, which yields a close-to-global optimal solution to the problem. The NMPC is not implementable in real time due to a large required computational effort. An explicit form of NMPC is developed, which approximates the NMPC solution to piece-wise constant solutions in the state space. The developed explicit MPC is able to reduce switching losses and minimise error. Linear control has been developed for comparison and developed anti wind-up removed limit cycles for velocity control. All results come from simulation on a developed non-linear model.

Pages, total:	174
Appendix:	7
Supplements:	1

By accepting the request from the fellow student who uploads the study group's project report in Digital Exam System, you confirm that all group members have participated in the project work, and thereby all members are collectively liable for the contents of the report. Furthermore, all group members confirm that the report does not include plagiarism.

PREFACE

This thesis documents a master thesis completed by group MCE4-1021 at the 10th semester at M.Sc education in Mechatronic Control Engineering at Aalborg University. The thesis concerns development and comparison of controllers for secondary control of discrete displacement cylinders.

The thesis is preceded with an introduction and state of the art analysis. This is followed by a short system description and is concluded by a chapter on the objective and approach of this thesis. The rest of the thesis is divided into three main parts. The first part concerns modelling of the system and the experimental validation. The second part describes control consideration, control design and simulated results. The third part concerns the comparison of the developed controllers based on tracking performance and energy efficiency and ends with a conclusion.

The initial thesis outline was aimed at testing developed controllers on a test setup readily available at Department of Energy Technology at Aalborg University. This included modelling and pressure control of a proportionally controlled symmetrical hydraulic cylinder to function as load for the discrete displacement cylinder. As the work progressed it became evident that the test setup was not suited for testing motion control without extensive work on the load side cylinder, which ultimately was outside the scope of this thesis. However, some work was completed on this part. While not directly relevant for this thesis, it is still included in the appendix for completeness. Appendix F describes the LQR pressure control developed, while appendix G describes the work done on improving the signal splitting between the two proportional valves controlling the load cylinder.

Reading guide
In this report the Vancouver-method is used when referring to sources. These references are listed at the end of the report with information such as author, title, edition and date. Figures and tables are numbered according to the chapter in which they occur. This means, that the first figure in chapter 3 has the caption "3.1" etc. The text, for explanation of the figures included, is located below the figures.

A .zip file containing the eMPC algorithm, the Simulink model and a compressed version of the report have been uploaded as supplement.

SUMMARY

An interest in reducing losses associated with high-force linear actuated hydraulic system has in this thesis led to investigating control of digital hydraulics. Three hydraulic topologies was investigated; Switch Control, Digital Flow Control Units and Secondary Control. Based on the state of the art analysis the objective of this thesis was to develop position and velocity control for a secondary controlled discrete displacement cylinder. A discrete displacement cylinder was readily available at Aalborg University and was used as test setup. The discussion following the state of the art analysis concluded that, while many secondary control investigations seeks an optimum between tracking performance and energy efficiency, none of the suggested implementations could claim optimality. Hence, the hypothesis of this thesis was to investigate if such an optimal control strategy for secondary controlled discrete displacement cylinder was possible. The optimal control strategy chosen, was Model Predictive Control.

To develop model based controllers a detailed mathematical model of the system was made. The developed model was validated by comparison to experimental results.

Some general control considerations of the system revealed the existence of limit cycles due to the discrete nature of the system, and further analysis revealed a relation between physical parameters of the system and the amplitude and frequency of the limit cycle. A fundamental difficulty of load holding was described due to a mismatch between available discrete force levels and the required force. Based on the control considerations the test setup showed to be impractical for position and velocity control as the mass of the load was too small. Furthermore, no real life application case study was readily available. Hence, an imagined application case was synthesised to be able to test the control strategies developed. The focus of the application case was to replicate difficult control problems found in real life applications more than describing a real life application.

A Non-linear Model Predictive Control scheme was developed. A reduced order discrete time model was derived by neglected pressure dynamics. A cost function penalising tracking error and switching losses was synthesised and physically motivated constraints where formulated. Results showed that it was possible to penalise switching losses, while maintaining good tracking response for both position and velocity control. However, it was concluded that Non-linear Model Predictive Control was not implementable due to the computational heavy optimisation problem.
An explicit Model Predictive Control scheme was developed in order to achieve the results obtained by the Non-linear Model predictive Control controller in real time. An algorithm to convert the Non-linear Model Predictive Control solution to an explicit solution was developed. Results from the

controller showed good tracking performance, while being able to reduce switching losses for both position and velocity control.
For comparison, a linear position and a linear velocity control was developed. Anti wind-up was investigated with the purpose of reducing or completely removing limit cycle behavior. The developed anti wind-up for the velocity control problem was to be able to remove limit cycles at the cost of introducing a small steady state error.

A comparison between Non-linear Model Predictive Control, explicit Model Predictive Control and linear control showed that it was possible to reduce the switching losses significantly with the two Model Predictive Controllers compared to the linear control, while obtaining similar or better tracking performance. Thus, with explicit Model Predictive Control it was possible to balance tracking performance and energy efficiency. Finally, it was concluded that position and velocity control was possible for a secondary controlled discrete displacement cylinder.

CONTENTS

CHAPTER 1

INTRODUCTION

Hydraulic systems are used in many applications where high force is needed, due to the high size-to-force ratio. A typical hydraulic system consists of a pump unit and a cylinder controlled by a proportional valve as illustrated in figure 1.1.

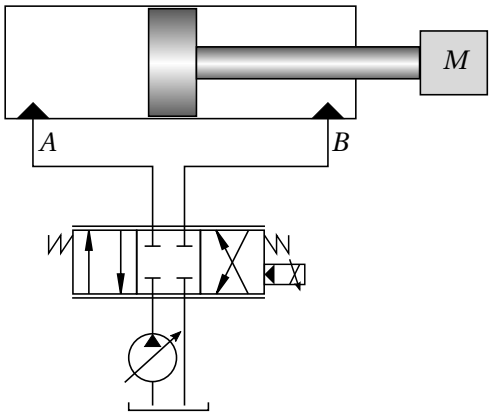


Figure 1.1. Illustration of a typical hydraulic system with a proportional valve.

By varying the spool position of the valve, an orifice between the the cylinder chamber and the pump or tank pressure line opens. The pressure difference across the valve orifice give rise to a flow through the orifice and into or out of the cylinder chamber. The meter-in and meter-out is not controlled independently by the proportional valve. The throttling and associated pressure drop across the orifice leads to energy losses. This becomes more significant when supply pressure is not close to the cylinder chamber pressure. The energy loss due to throttling becomes significant as the power of the hydraulic application increases. Examples of such applications can be seen in the off-shore industry, where there is potential for large savings in fuel costs for hydraulic applications.

Digital fluid power is a concept that has been subjected to increasing research. The focus of the research has been on using digital hydraulics to design energy efficient, low cost and robust systems. Thus, the concept is interesting in relation to high-power off-shore applications in order to increase the efficiency and thereby save money on fuel. The idea of using digital hydraulics is to introduce simple on/off valves, which can replace traditional proportional valves. Digital refers to the discrete valued nature of the on/off valves used for control of the system. According to [1] two configurations

of digital fluid power exist, namely one based on parallel connection of on-off valves and another based on switching technologies. In [2] and [3] a third configuration is presented; secondary control. In figure 1.2 an illustration of how each of the three configurations can be used to replace the traditional proportional valve is shown.

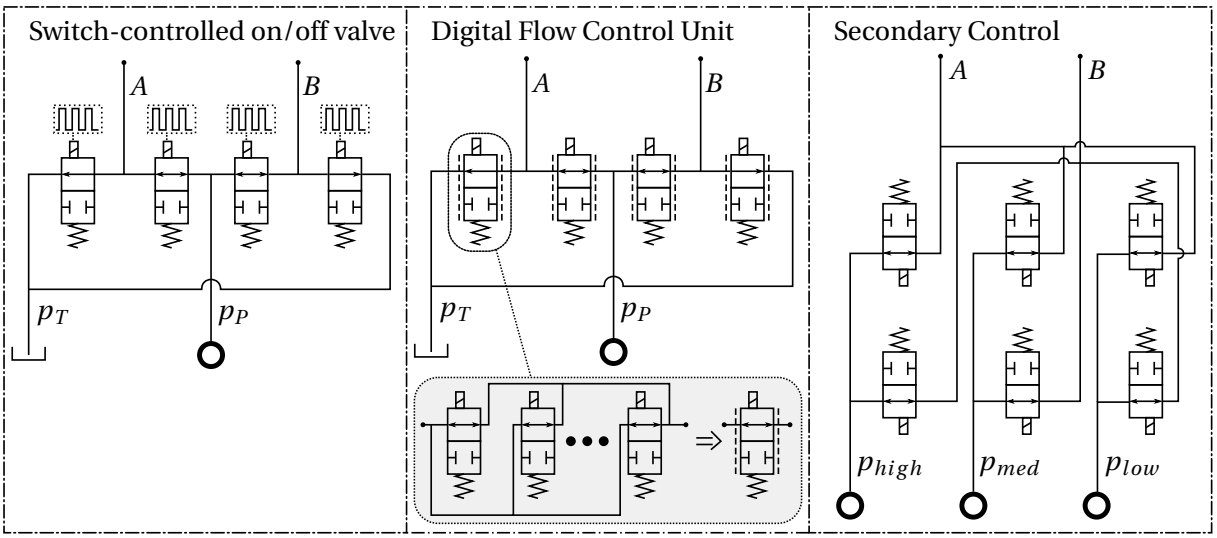


Figure 1.2. Illutstration of different digital valve configurations.

The switching controlled configuration is shown in 1.2 to the left. Switching technologies uses fast continuous switching valves, where the output is controlled by e.g. pulse width modulation (PWM). The parallel connected system consist of a number of parallel connected on/off valves, often referred to as a Digital Flow Control Unit (DFCU). The symbol of a DFCU is similar to the symbol of a proportional valve, but the lines indicating proportionality are dashed, to point out that the opening area is discrete in nature. The third configuration used for secondary control is based on a single or more valves in parallel, all operated as a single on/off valve. The secondary control configuration is not ideal for fast continuous switching of the valve, but instead the valve is operated as a open or close state.

In the following chapter a detailed state of the art analysis of the three configurations and some relevant applications are presented.

CHAPTER 2

STATE OF THE ART

In this chapter a state of the art analysis of the three digital fluid power topologies switch control, digital flow control units and secondary control are presented. For each of the topologies one or more applications are presented. By the end of the chapter a discussion of the conclusions drawn from the state of the art analysis is made. The goal is to identify the preceding research on the area and what still needs to be investigated in order to identify the scope of this thesis.

2.1 Switch Control

The switch control valve topology is similar to the well known switch mode control of electric actuators, such as switch control of an inverter for an electric motor. Similar to the electric switch mode control, the switch control of digital valves is used to control the average output of the switch by e.g. PWM control. The switch control of digital valves control the average flow area of the orifice. In figure 2.1, 2.3 and 2.2 three switch control systems are given, a hydraulic buck converter, an elementary valve switch system and a pneumatic clutch.

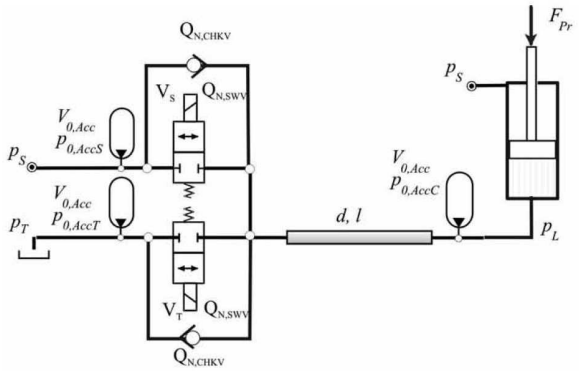


Figure 2.1. Sketch of hydraulic buck converter [4].

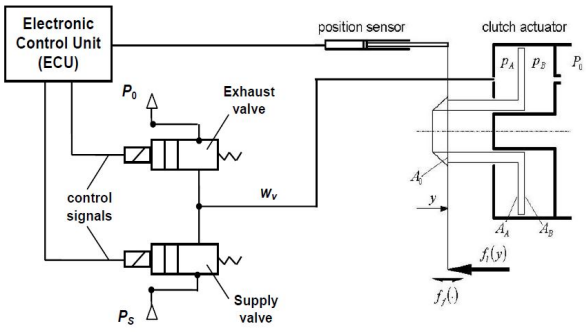


Figure 2.2. Sketch of pneumatic clutch [5].

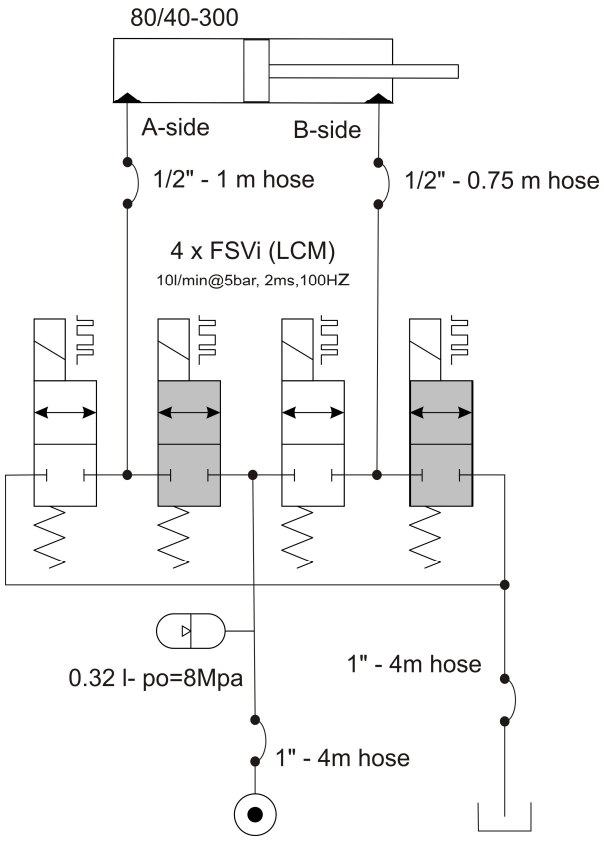


Figure 2.3. Configuration with switch controlled digital valves [6].

In [4] a compact hydraulic buck converter, corresponding to an electric buck converter, is investigated and developed. The hydraulic buck converter is designed for actuation of mobile outdoor robotic applications. It uses switch controlled on/off valves configured as sketched in figure 2.1. An advantage of the design is the possibility to recuperate energy and great energy reduction is achieved in comparison with a proportional valve.

In [6] motion control of a hydraulic cylinder is performed with PWM control of digital valves, with the setup shown in figure 2.3. In contrast to a conventional proportional valve the meter-in and meter-out are controlled independently. Different PWM frequencies are tested and controllability and dynamic behaviour is investigated. Increased frequency and thereby faster switching of the valves decreases pressure pulsations and yields better controllability, but does also increase the acoustic noise level. Finally it is concluded that the system provides comparable control performance as a proportional valve in terms of response dynamics and precision.

The pneumatic clutch shown in figure 2.2 is investigated in [7] and [5]. The on/off valves are either fully open or fully closed and are controlled by switching. In [7] a control law is designed using backstepping theory and is seen to fulfil the accuracy and response requirements. The robustness of the controller is investigated by adding white noise to the position measurements and robustness to noise is seen. An issue however with the back-stepping method is that it lack the ability to handle constraints on actuator position. In [5] a solution to the control problem of the pneumatic clutch is to use optimal control which can handle both state and input constraints. The optimisation based control method proposed is Model Predictive Control (MPC). Using MPC for reference tracking

control problem of a non-linear constrained system with a quantised input makes it possible, to penalise both input and system states via weight constants in the cost function. Due to fast dynamics in the clutch actuator and a small sampling time an explicit approximate solution is utilised.

2.2 Digital Flow Control Units

The DFCU can be seen as a proportional valve, but with discrete nature, where the flow area is the sum of the open valves in the DFCU, as illustrated in figure 2.4. The flow can be controlled by the combination of the open valves in the DFCU and thereby controlling the flow by throttling, similar to a proportional valve. The resolution of the opening area is dependent on the number of valves in parallel and the coding scheme, which refers to the size of the individual valves. Different coding schemes exists, with the highest resolution achieved with binary coding, where the second valve is twice the size of the first valve, the third valve is twice the size of the second valve and so forth. This gives the size ratio 1:2:4:8 etc. Binary coding gives $2^n - 1$ opening combinations, where n denotes the number of valves in parallel in the DFCU. An essential difference between the switching valves and the DFCU is that no switching is required to maintain the output of the DFCU, as this is dependent of the valve states.

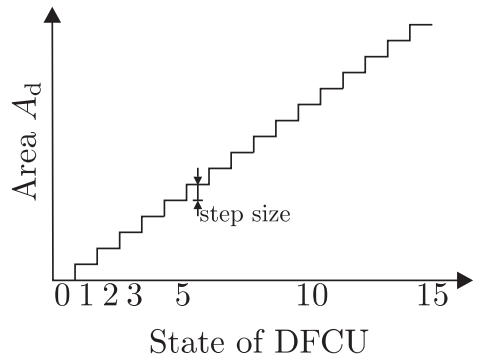


Figure 2.4. Area of DFCU as function of valve states [8].

Binary coding requires that valves are closed and opened simultaneously, which for non-ideal valves can cause pressure peaks and pressure pulsation, due to difference in switching time. The problem is investigated in [9] where a binary coding scheme is compared with a Fibonacci coding scheme. In the Fibonacci coding scheme the ratio of the valve size is 1:1:2:3:5 etc and is a combination of binary coding and Pulse Number Modulation (PNM), where all valves have same size. The advantage with PNM is that valves are either opened or closed simultaneously but not both, but the disadvantage is that it requires a large number of valves to obtain a resolution equal to binary coding. The article can not conclude that the Fibonacci coding scheme limit pressure peaks compared to binary coding.

2.2.1 Resistive Control

The general idea of resistance control as presented in [10], [2] and [11] is to utilize the many force levels of a multi-chamber cylinder for velocity control. With a DFCU the force level is selected slightly above the force required for the wanted acceleration. Velocity control is then achieved by throttling the flow. This way, the full displacement flow is throttled, but at a much better matching between supply and load pressure, hence reducing throttling losses. The system under investigation was a three chamber cylinder with two pressure lines, a pump pressure line and a pressurised tank line.

Switching between pressure modes is penalised with a hysteresis term and a maximum switching frequency. The control strategy was tested with both normal and overrunning loads. The test setup is shown in figure 2.5.

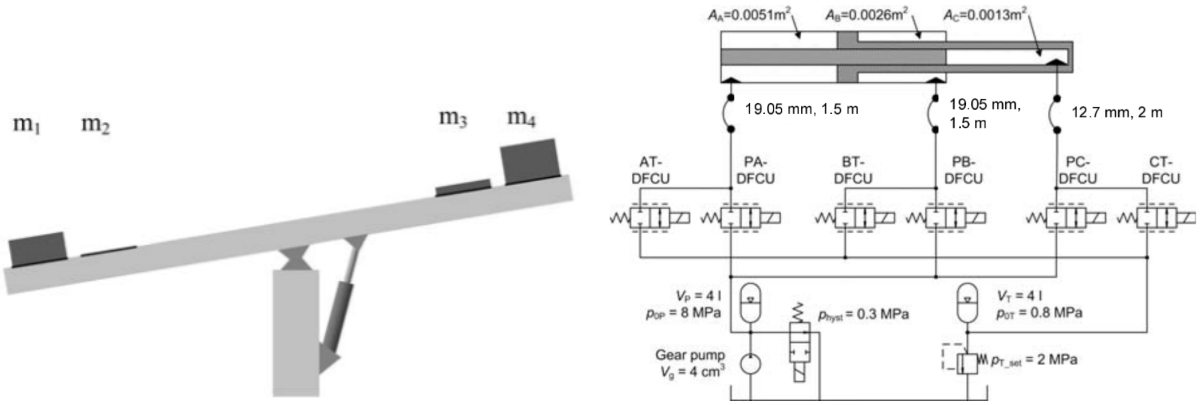


Figure 2.5. To the left the test setup and to the right the corresponding hydraulic setup used to test resistive control in [10].

With the configuration the meter-in and meter-out are controlled independently. The authors of the articles state that even though throttling is used to control the flow, energy efficiency is increased by introducing digital valves instead of proportional valves. In [10] the energy loss of the system is compared with a traditional load sensing proportional valve and 33% average energy loss reduction is seen.

2.3 Secondary Control

Secondary control consists in general of directly connecting a hydraulic actuator and accumulator to a constant pressure line. The hydraulic accumulator is then used as short term energy storage, for when the hydraulic actuator supplies energy to the high pressure line. Long term bi-directional power flow is then possible, if the pump supplying the high pressure line is bi-directional. Thus, the benefits of secondary control are primarily energy recuperation from the hydraulic actuator, which could be both a pump/motor drive or linear drive. This thesis focus on secondary control of a linear drive, in this case a multi chamber cylinder.

Secondary control of a multi chamber cylinder involves connecting actuator ports directly to pressure lines through a number of on-off valves. The valves are ideally sized such that throttless flow is achieved under normal working conditions. Parallel connections between pressure line and actuator enable a multitude of force levels and allows the user to generate and store energy, e.g. recuperate energy from braking by controlling the system as a hydraulic transformer. The direct connection between actuator volume and pressure line leads to a discrete output. Thus, for low mass systems velocity can be difficult to control, due to large acceleration caused by the step like input. In [1] it is stated that secondary control is one of the most energy efficient ways, to control a hydraulic cylinder from constant pressure lines.

2.3.1 Power Take-Off Systems

Secondary control of digital hydraulics has found use in Wave Energy Converters (WEC) as hydraulic transmissions in Power Take-Off (PTO) systems. The WaveStar prototype employs such a PTO system, which is thorough described in [3] and [8]. The main principle of the WaveStar PTO system is based on boyant point bodies resting on the water surface. The float is hinged on a lever arm, providing a rigid coupling to the sea bed. Energy is then extracted by applying a force to the point absorber to resist motion caused by waves. A concept drawing can be seen in 2.6

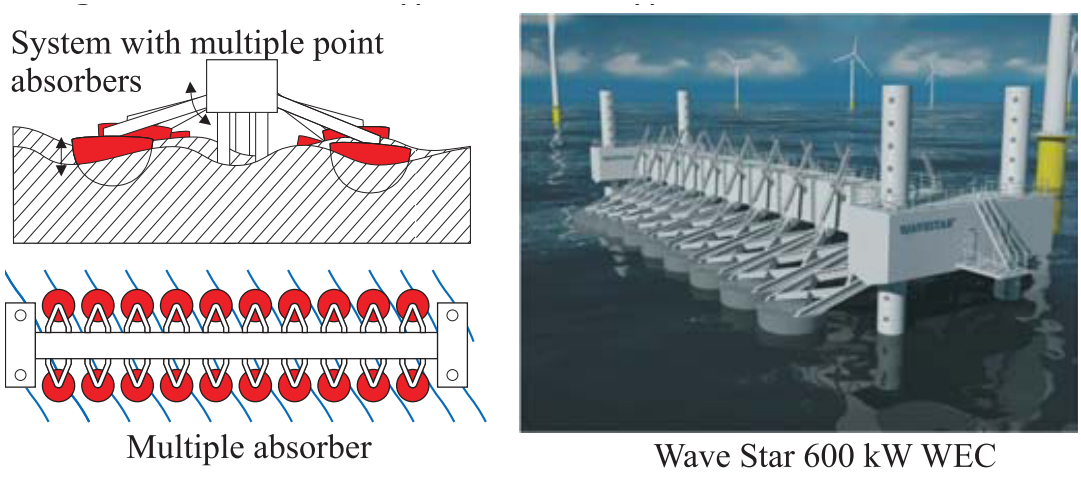


Figure 2.6. WaveStar concept as presented in [12]

The PTO system is based on a three chamber cylinder with three pressure lines, yielding a total of 27 force levels distributed symmetrically around zero. The many force levels allows the PTO system to closely track a force reference. Both resistive and reactive control, where the PTO emulates a mass-spring system, was tested in [3], utilising the high-efficient four quadrant control made possible by the digital hydraulic components. The power supplied by the cylinder to the common rail pressure is then converted to electrical energy by hydraulic motors and generators. A full scale WaveStar PTO prototype is located at Aalborg University. In [12] a force control was implemented on the WaveStar PTO prototype with a force selection algorithm based on force error minimisation and a maximum frequency of switching. A second selection algorithm included energy costs of switching and a hysteresis band, enabling user to compromise between tracking error and energy cost. The latter is the selection algorithm currently implemented on the WaveStar PTO prototype. However, no work has been done on analysing PTO-efficiency versus force tracking error to optimise efficiency as a function of tracking error.

Pelamis is another WEC project, which is presented in [13]. It is based on floating linked structures aligned perpendicular to the wave direction. Energy from the relative rotational motion of the sections due to wave motion, is then extracted using a hydraulic PTO system. The PTO system is based on identical pairs of differential hydraulic cylinders connected to a high and low-pressure line, yielding eight force combinations. The cylinder pair apply a torque to resist the rotational movement between the sections. Cylinders are connected to high and low pressure accumulators via rectifying check valves. The high and low pressure lines are then connected to a hydraulic pump-motor, enabling power to be extracted from the pressure lines.

Pressure Peak Minimisation

In [3], [8] and [14] the WaveStar PTO prototype described earlier is investigated. As previous described the three chamber cylinder with three pressure lines yields 27 force levels. The 27 force levels and a sketch of the multi-chamber cylinder with its three three pressure lines is shown in figure 2.7. The multi-chamber cylinder is connected to the three pressure lines through a digital valve manifold containing 9 valve packs. Each of the 9 valve packs contains a number of on/off valves in parallel. Thus, it could be seen as a DFCU.

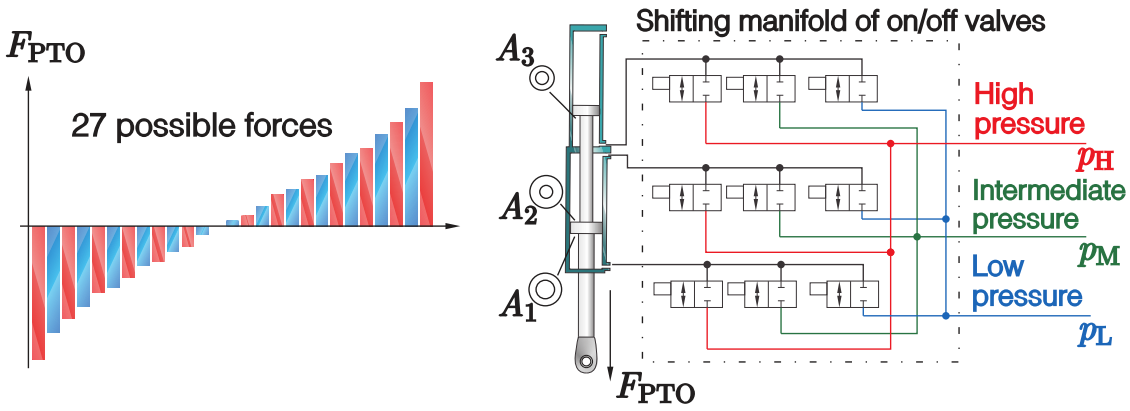


Figure 2.7. Illustration of the test setup used for Wave Star and possible force levels of the multi-chamber cylinder [14].

Operating the 9 valve packs as single on/off valves, result in fast shift between pressure levels, which introduce pressure peaks and oscillations in the actuator chamber and transmission lines. In [8] pressure peaks are evaluated as function of the valve switching time. Faster valve switching is seen to minimise the pressure peaks, but also yield larger pressure oscillations. In order to minimize pressure oscillations the valve opening characteristic is considered. In [15] a valve opening characteristic based on a desired pressure trajectory is proposed. The valve opening is an open-loop control strategy, which is calculated based on a theoretical model of the system. Hence, a pressure trajectory is given as input and a model based valve reference as output. By shaping the valve opening characteristic based on a third or fifth order pressure trajectory the pressure oscillations are reduced compared to a linear valve opening characteristic. The concept is investigated further in [16], where the energy loss associated with the valve shifting is also considered and a sinusoidal pressure trajectory is tested. The experimental setup in both [15] and [16] is made with proportional valves instead of digital valves. However, in [14] the calculated valve opening characteristic is simulated with digital valves and similar results are obtained.

2.3.2 Motion Control

In [17] velocity control for an excavator arm was attempted with PI-control in series with a force step selection algorithm and the focus of their work is the latter part. The hydraulic test setup was a four chamber cylinder connected to a three level common pressure rail, generating 81 force steps. Two strategies were tested. One selection algorithm is based on a force error minimisation with a hysteresis-like switching penalty. A second strategy used a score-based selection, where switching between force steps with large difference was penalised further in order to obtain a more

smooth velocity. Difficulties and further research where identified in the differences in hydraulic capacitances, leading to force transients counteracting the desired piston movement.

A single commercial product utilising digital hydraulics for motion control is available in NorrDigiTM, developed by NorrHydro Oy. A single research article was published in [18] and other information is available on the webpage. The NorrDigi is based on a multi-chamber cylinder connected to a common three level pressure rail through a digital valve manifold and is aimed mainly at mobile applications, e.g. cranes, excavators, thus reducing fuel costs, and increasing reliability. An illustration of a conceptual implementation of NorrDigi can be seen in figure 2.8.

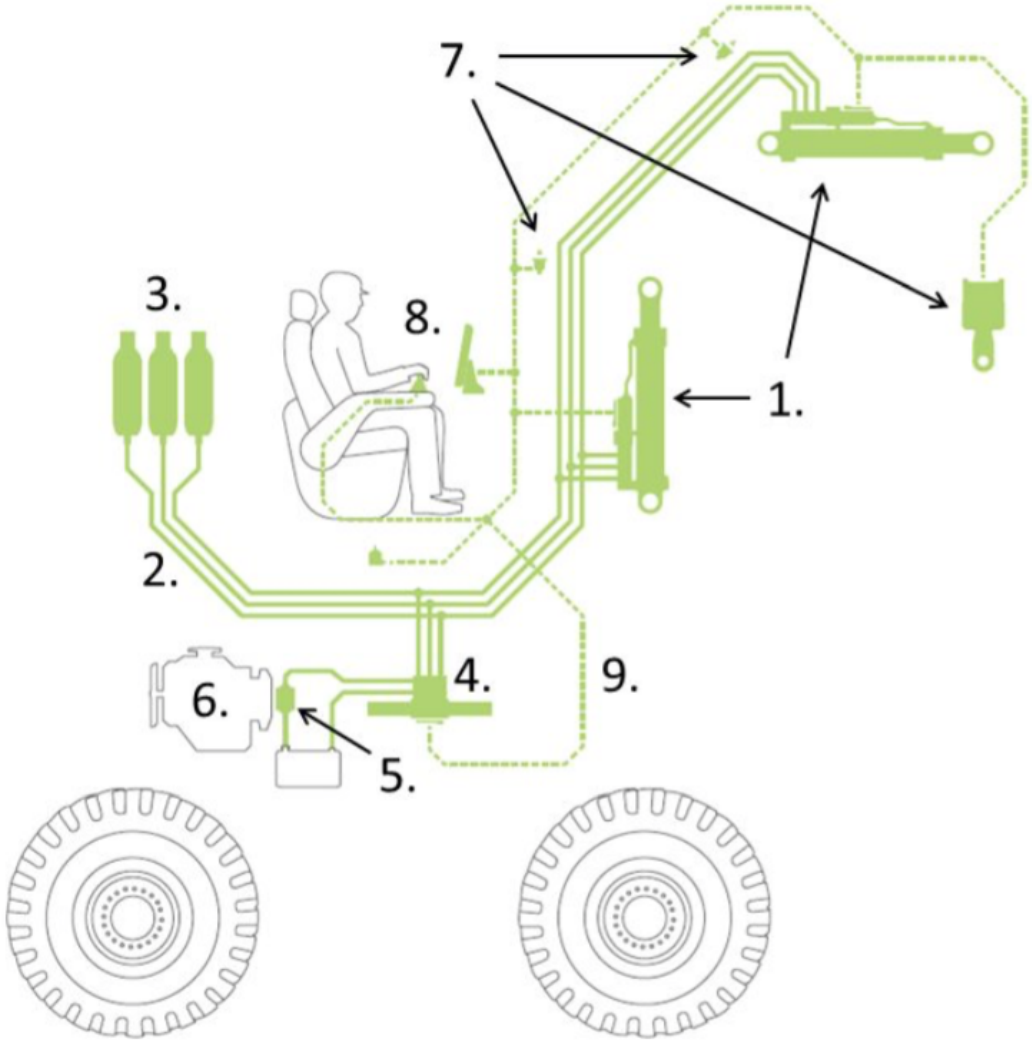


Figure 2.8. NorrDigi implementation on a mobile work machinery as presented in [18].

The three level common pressure rail (2) connects the multi chamber cylinders with integrated valve packs (1) to the accumulators (3) and an energy level management unit, which aims to stabilise pressure, and enable power recovery. The system works with commercially available pumps (5), such as fixed displacement pumps. The system is interfaced with CAN-bus (9). Thus, NorrDigi is a highly integrated product. The datasheet on control software of NorrDigi claims to calculate optimal control signals, but no information is available for which cost is minimised.

2.4 Discussion

In this state of the art analysis three different digital fluid power topologies using on/off valves have been presented.

The investigated applications using switch controlled on/off valves, generally showed good performance with respect to energy efficiency, dynamic response and accuracy. A disadvantage of one of the investigated applications was a compromise between pressure pulsations and the acoustic noise level.

Resistive control of the DFCU topology used in combination with a multi-chamber cylinder, showed good results with respect to energy efficiency. The topology was compared with a traditional load sensing proportional valve system and 33% average energy reduction was seen.

The secondary control topology has found use in both WECs and motion control of multi-chamber cylinders. On the WaveStar PTO prototype system two control algorithms have been tested. One based on force error minimisation and a maximum frequency of switching, and a second including energy costs of switching and a hysteresis band, enabling the user to compromise between tracking error and energy cost.
Another application perform velocity control on an excavator arm with a PI-controller in series with a force step selection algorithm. The focus of the implemented control algorithm is only on reference tracking and no attention is given to the energy efficiency of the system.
The product NorrDigiTM, which is a commercial product, have implemented some kind of optimal control, which could include an optimum between energy efficiency and tracking performance. However, product information is limited, so it is not possible to determine the implemented control algorithm.
Common for secondary control is that high energy efficiency is achieved because the generated energy from the actuator can be recovered.

Based on the state of the art analysis of the three topologies, major disadvantages has not been determined for any of the topologies. Thus, all three topologies could be interesting to investigate further. However, in order to narrow the area of interest and because the test setup used in [14], [8] and [3] is available for the project group, the focus of this thesis will be on secondary control of multi-chamber cylinders actuated by a digital manifold consisting of nine DFCUs.

Common for the research on secondary control of multi-chamber cylinders is that a control algorithm taking the tracking error and efficiency into account has not been thorough investigated. So the idea of using optimal control seems appealing. As mentioned under switching control the optimal control method MPC is proposed in [5]. The method is able to successfully optimise the control output based on system inputs and states. This method however is utilised on a switching topology system. In [19] promising results using MPC have been shown on systems with a low rate of change in the control output, thus making MPC an interesting prospect for further investigation in this thesis.

In the next chapter a brief description of the Wave Star test setup from [14], with its secondary controlled multi-chamber cylinder, is presented. After this a detailed description of the objective and approach for the thesis.

CHAPTER 3

DESCRIPTION OF SYSTEM

As mention in section 2.4 on the facing page a test setup with a secondary controlled multi-chamber cylinder is available for the project, thus this system is here briefly described. The test setup investigated is the full scale PTO unit of the 'Wave Star' Wave Energy Converter (WEC) test bench located at Aalborg University [3]. The test bench is shown in figure 3.1. The test setup consist of two horizontally opposing cylinders connected with a low mass sleigh. On one side a 420kN multi-chamber cylinder with five chambers operated as three. The cylinder is actuated by secondary control by switching between three constant pressure lines in a digital hydraulic manifold. This gives the cylinder a discrete flow input, thus the entire secondary controlled multi-chamber cylinder is further on called a Discrete Displacement Cylinder(DDC) [14]. The other side is a 840kN standard proportionally controlled symmetric hydraulic cylinder acting as a load cylinder. The test setup is mainly being used to test a Power Take Off (PTO) system for a WEC application, where the load cylinder has been designed with a motion controller in order to emulate waves.

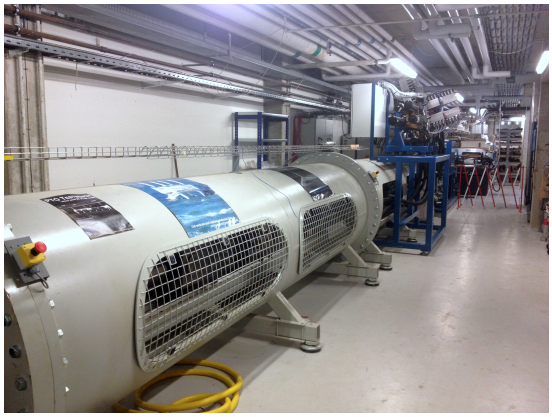


Figure 3.1. The WaveStar PTO test bench.

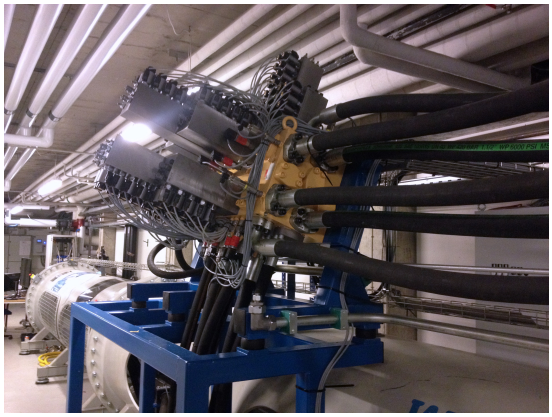


Figure 3.2. The valve manifold of the PTO test bench.

On figure 3.3 a sketch of the entire test setup is illustrated.

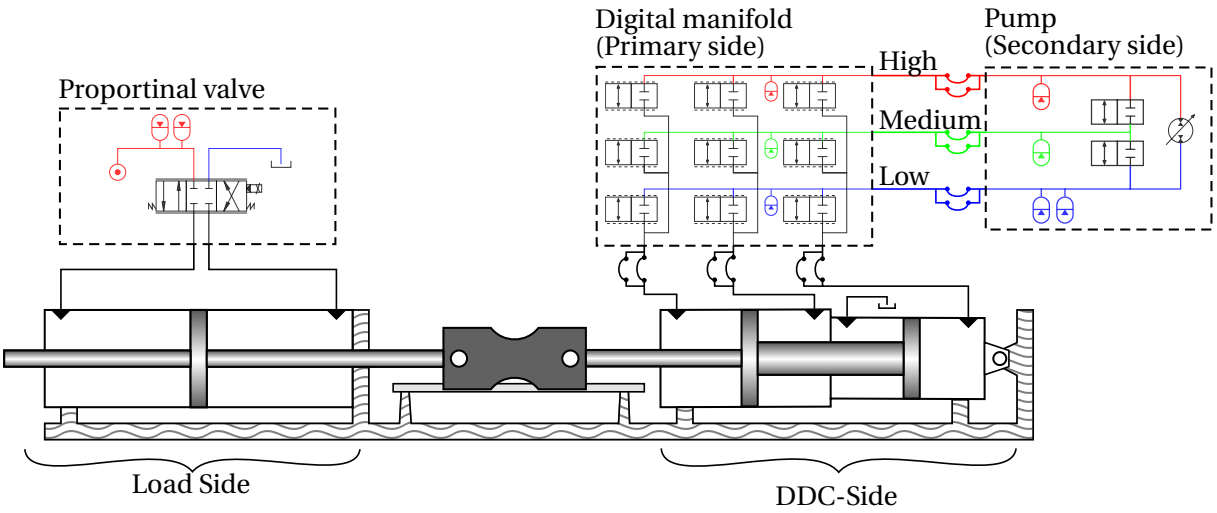


Figure 3.3. Sketch of test setup.

3.1 Discrete Displacement Cylinder

The DDC system is divided into a primary and secondary side. The primary side consist of a multi-chamber cylinder with five chambers. Two chambers are operated in parallel and one is connected to air at atmospheric pressure. This yields effectively three chambers, which are sized to have an approximately symmetrical force distribution. The total stroke length of the cylinder is 2m, but is mainly operating in ±0.5m from center position. The manifold of the cylinder have nine on/off-valve units, which can be seen as DFCUs, connecting the cylinder chambers to three different pressure levels. This gives a force distribution of 27 different discrete force levels as seen in figure 3.4. Each of the DFCUs consist of 8-10-18 standard off-the-shelf Bucher on/off-valves sized to match the corresponding flow area of each chamber. Each pressure line are fitted with gas-loaded hydraulic piston accumulators to reduce peaks in line pressure levels due to switching. The valve manifold an pressure lines are seen in figure 3.2.

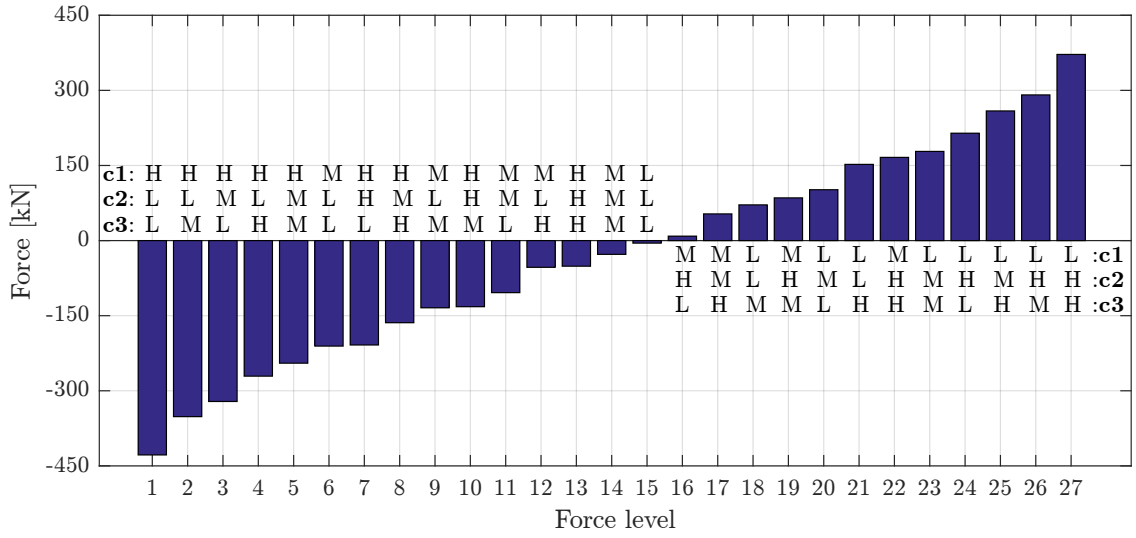


Figure 3.4. Force distribution of the 27 different force levels. The letters denotes the pressure levels in the chambers, where $H = 200$ bar, $M = 107.5$ bar and $L = 20$ bar.

The secondary side of the DDC system is designed to supply three pressure levels and enable secondary control. To maintain the high side pressure level the system is connected to a 250cc close-circuit variable displacement pump/motor between the high and low pressure lines. The mid-pressure is then obtained by two proportional valves from the high and low pressure lines respectively. Each line have a 25L accumulator for short-term energy storage. One source of issue regarding the test setup design is that, in order to allow easy access to internal components, the valve manifold has been mounted externally on the tube. This causes transmission lines of 2-4m from manifold to cylinder [3].

3.2 Load Side

The load side cylinder used in the setup have stroke length of 3m, with center aligned with center of the DDC. It is able to deliver a force of 840kN. The load cylinder consist of two chambers with equal piston areas. The flow to the cylinder is controlled by two 4/3 proportional valves in parallel. A Parker D111FP valve with a rated flow of 1000 L/min at 5 bar and overlap of 10%. The overlap around center position is compensated by the other valve in parallel, a Moog D634 with a rated flow of 100 L/min at 35 bar. The supply pressure is generated by two variable displacement pumps of 125cc and 250cc. The supply pressure is furthermore fitted with two 28L accumulators.

CHAPTER 4

OBJECTIVE AND APPROACH

The state of the art analysis revealed, that limited research has been made on tracking performance versus energy efficiency for secondary controlled multi-chamber cylinders. The hypothesis is that an optimum between tracking performance and energy efficiency can be determined, with the aid of an optimal control scheme. As presented in the state of the art analysis promising results have been obtained with the optimal control scheme, Model Predictive Control, for two different systems with quantised inputs. However, MPC has not been attempted on secondary controlled multi-chamber cylinder systems. Thus, the main objective of this project will be to design a MPC for the Discrete Displacement Cylinder used on the WaveStar PTO prototype test setup. The mass of the sleigh connecting the two cylinders in this test bench is relatively low. In order to perform motion control of secondary controlled systems a high mass to force ratio is preferable. Due to the low mass of the sleigh the developed control algorithms will not be tested on the test setup, but the test setup will instead be used to validate a model of the DDC. The performance of the controllers are then evaluated based on simulations with the validated DDC on an application with a larger mass.
In order to design a NMPC for the system, the following approach will be used.

A detailed model of the entire test setup will be made. This include the equations describing the multi-chamber cylinder and the digital valve manifold connecting the cylinder to three pressure lines, and the equations describing the load cylinder. The equations are implemented in MATLAB Simulink in order to obtain a non-linear model of the whole system.

The non-linear model is verified in order to use the model for control development and later for control evaluation. The model is verified by comparing it with experimental data from the test setup. Soft parameters are identified and adjusted to give the best similarity between model and experimental data.

Requirements for controller design and criteria for which the controllers will be evaluated. In order to state the requirements for the controller to the multi-chamber system, the losses of the system associated with shifting from one force level to another will be investigated for the purpose of including it in the MPC. Furthermore the calculation of the tracking error is presented likewise in order to include it in the MPC.

Development of controllers based on the stated requirements. In the design of the MPC the equations expressing the losses and tracking error will be included in the cost function, in order for the MPC to optimise based on this two criteria. The main objective is to design an MPC for the system. However, a PI controller will also be design in order to compare the MPC with a

simple controller. Simple in the sense of few design parameters. Even thought the PI controller is a simple controller in terms of design parameters, it require some thoughts to make it work for the system, due to the quantized input. The PI-controller is implemented together with a force selecting algorithm.

Implementation and evaluation of controllers. The designed controllers are implemented in the non-linear model in MATLAB Simulink in order to evaluate control performance. Since the objective is to investigate the tracking performance and the energy efficiency, each controller is evaluated based on this criteria.

Controller comparison with regard to tracking performance and energy efficiency. The comparison will be used to determine which of the controllers, that are most suitable for the system. Furthermore, the robustness of the controllers to changes in model parameters are evaluated.

Linearisation of the load side system and verification of it based on the non-linear model, can be found in the appendix section F. Furthermore, development of an LQR controller for the load side system can be found in the appendix section. Ideally a force tracking reference should be designed to emulate friction and acceleration of the mass, in order to test and verify the controllers on the PTO prototype test bench. The reference could be developed based on a model of the system. However, it is outside the scope of this thesis and will therefore not be developed.

Part I

Modelling & Validation

CHAPTER 5

SYSTEM MODELLING

In this chapter a thorough description of all equations and parameters used to model the system described in section 3 are presented. First the DDC model is described including the manifold valve characteristics and transmission line dynamics. Next the equations regarding the load-side cylinder are presented. Furthermore, the system mechanics are described with a friction model and Newtons 2. law. An overview of the entire model is presented in section 5.4. Lastly, the energy losses associated with switching between the discrete force levels and the throttling losses of the manifold are presented.

5.1 Modelling of Discrete Displacement Cylinder

In this section a dynamic model of the DDC is presented. The DDC is illustrated in figure 5.1. Control of the constant pressure lines from the pump-side is outside the scope of this project. Thus, it is assumed that the pressure lines delivers three constant pressures for the DDC. It is possible to set the pressures in the pressure lines manually, but for the rest of this thesis the line pressures used are; p_H = 200 bar, p_M = 107.5 bar and p_L = 20 bar. A common notation for the line pressure is p_S=[p_H p_M p_L].

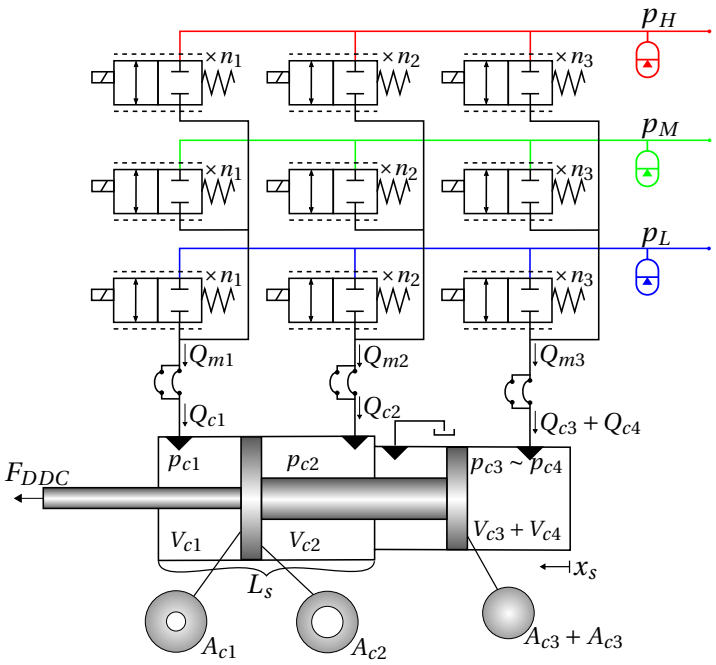


Figure 5.1. Hydraulic schematic of the Discrete Displacement Cylinder with a multi-chamber cylinder and a valve-manifold consisting of 9 Digital Control Flow Units.

As presented earlier the cylinder is a 5 chamber cylinder, where only 4 chambers are used, with chamber 3 and 4 operated in parallel. This means that for the schematic in figure 5.1 the sum of the two chambers are illustrated as one chamber. Physically the flow out of the third column of valves is split out into two individual chambers. Thus, the pressure dynamic in each of the 4 chambers are described by the continuity equation as

$$\dot{p}_{c1} = \frac{\beta_e(p_{c1})}{V_{0c1} - A_{c1}x_s}(Q_{c1} + A_{c1}\dot{x}_s) \tag{5.1}$$

$$\dot{p}_{c2} = \frac{\beta_e(p_{c2})}{V_{0c2} + A_{c2}x_s}(Q_{c2} - A_{c2}\dot{x}_s) \tag{5.2}$$

$$\dot{p}_{c3} = \frac{\beta_e(p_{c3})}{V_{0c3} + A_{c3}x_s}(Q_{c3} - A_{c3}\dot{x}_s) \tag{5.3}$$

$$\dot{p}_{c4} = \frac{\beta_e(p_{c4})}{V_{0c4} + A_{c4}x_s}(Q_{c4} - A_{c4}\dot{x}_s) \tag{5.4}$$

Where $\beta_e(p_{ci})$ and V_{0ci} are the effective oil bulk modulus and initial volume respectively of the i'th cylinder chamber. Q_{ci} is the total flow into the i'th chamber.
The effective oil bulk modulus is calculated using Lee's model given as [20]

$$\beta_e(p) = 0.5\beta_{max}\log\left[100\left(0.9\frac{p}{p_{max}} + 0.03\right)\right] \tag{5.5}$$

where $p_{max} = 280$bar is the maximum load pressure for which Lee's model is valid and β_{max} is the fluid bulk modulus at p_{max}. The fluid bulk modulus is chosen to be $\beta_{max} = 18000$bar [20]. This value however is a soft parameter dependent on many not directly measurable values, like the amount of air in the oil, and it will therefore be re-evaluated in the validation of the system.

5.1.1 Manifold

The manifold consist of nine valve units with several on/off valves in parallel, which can be seen and operated as DFCUs. As illustrated on figure 5.1, each chamber of the cylinder is connected to a

column of three valves connecting it to each of the pressure lines. A chamber can only be connected to one pressure line a time, thus only one DFCUs is opened in each column. Therefore there are always a maximum of three DFCUs open at a time .

The flows through the nine DFCUs are modelled using the orifice equation. Each of the DFCUs consist of several on/off valves in parallel. Thus, the area of the orifice is dependent on the number of on/off valves which are open. In order to take the number of on/off valves in parallel into account, the orifice equation is scaled with a gain corresponding to the number of valves in the DFCU. With the orifice equation the flow from A to B is given as

$$Q_{AB} = n_i A_0 c_d u \sqrt{\frac{2}{\rho}(p_A - p_B)} \text{sgn}(p_A - p_B) = n_i k_v \sqrt{(p_A - p_B)} \text{sgn}(p_A - p_B) \tag{5.6}$$

where A_0 is the opening area of 1 on/off valve, c_d is the discharge coefficient, u is a normalised input between 0 and 1, ρ is the oil density, and p_A and p_B are the pressure before and after the DFCU respectively. n_i is the number of valves in the DFCU for the i'th chamber, and $n_1 = 18$, $n_2 = 10$ and $n_3 = 8$. k_v is the valve gain. From figure 5.2, $k_v \sim 1 \cdot 10^{-6} \left[\frac{\text{m}^3/\text{s}}{\sqrt{\text{Pa}}}\right]$ for flow direction 2-1, which is supply side to load side.

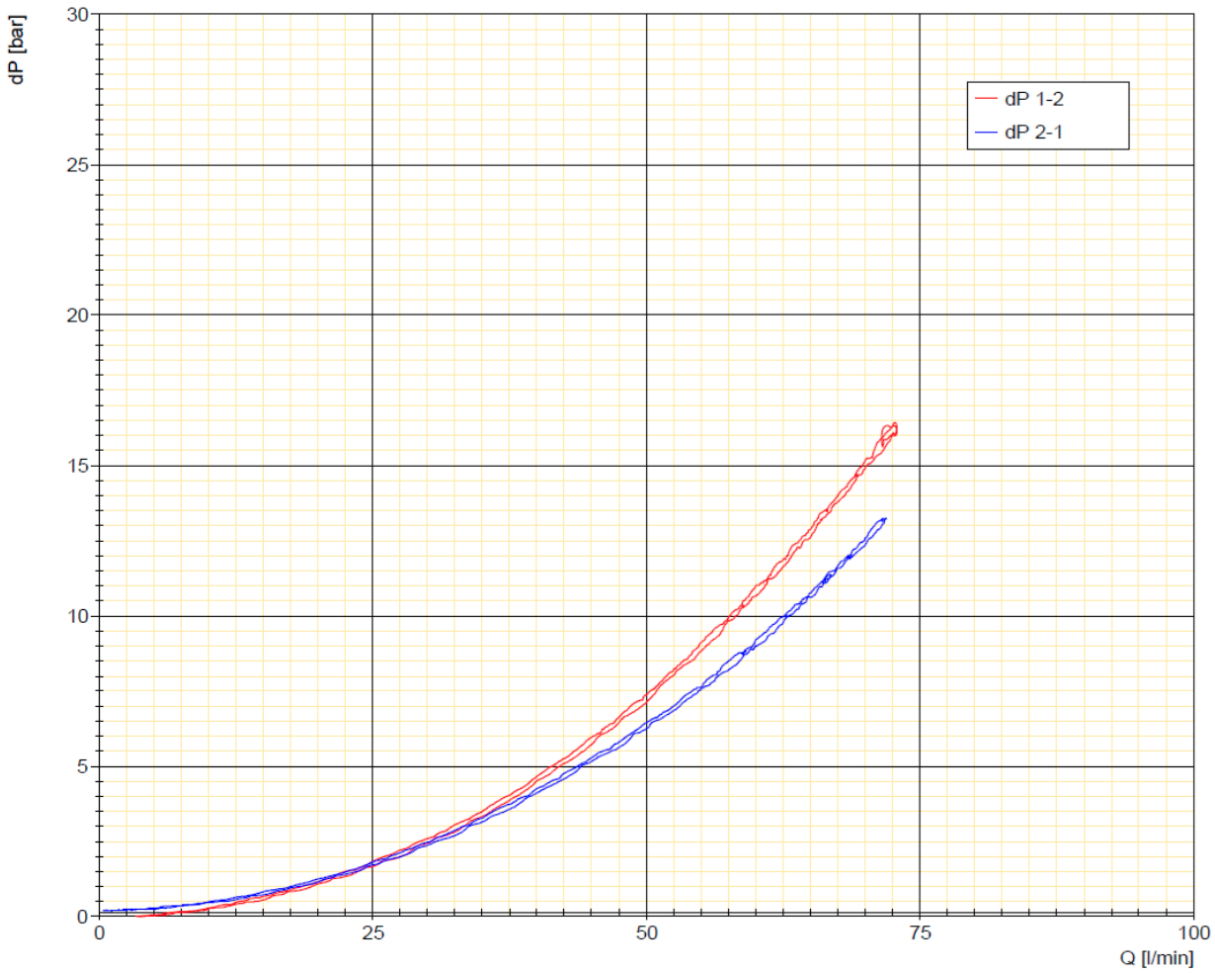


Figure 5.2. Valve gain characteristics. 1-2 and 2-1 refers to flow direction.[21]

As seen on figure 5.1 the large DFCUs with n_1 valves is connected to chamber 1, the DFCU with n_2 is connected to chamber 2 and finally the DFCU with n_3 valves is connected to the parallel operated

chambers 3 and 4. This means that the third flow out of the manifold, Q_{m3}, is split out to a flow for each chamber; Q_{c3} and Q_{c4}.

$$Q_{m3} = Q_{c3} + Q_{c4} \tag{5.7}$$

The valve dynamic is described based on pressure curves supplied by the valve manufacturer. It is modelled as a 9ms delay on the input and a second order output dynamic given as

$$\frac{u}{u^*} = \frac{\omega_{nv}^2}{s^2 + 2\zeta_v \omega_{nv} s + \omega_{nv}^2} \tag{5.8}$$

where $\omega_{nv} = 630$ rad/s is the eigenfrequency of the valve and $\zeta_v = 0.95$ is the damping of the valve, determined approximately from figure 5.3. It is seen that the pressure reaches steady state after approximately 20ms.

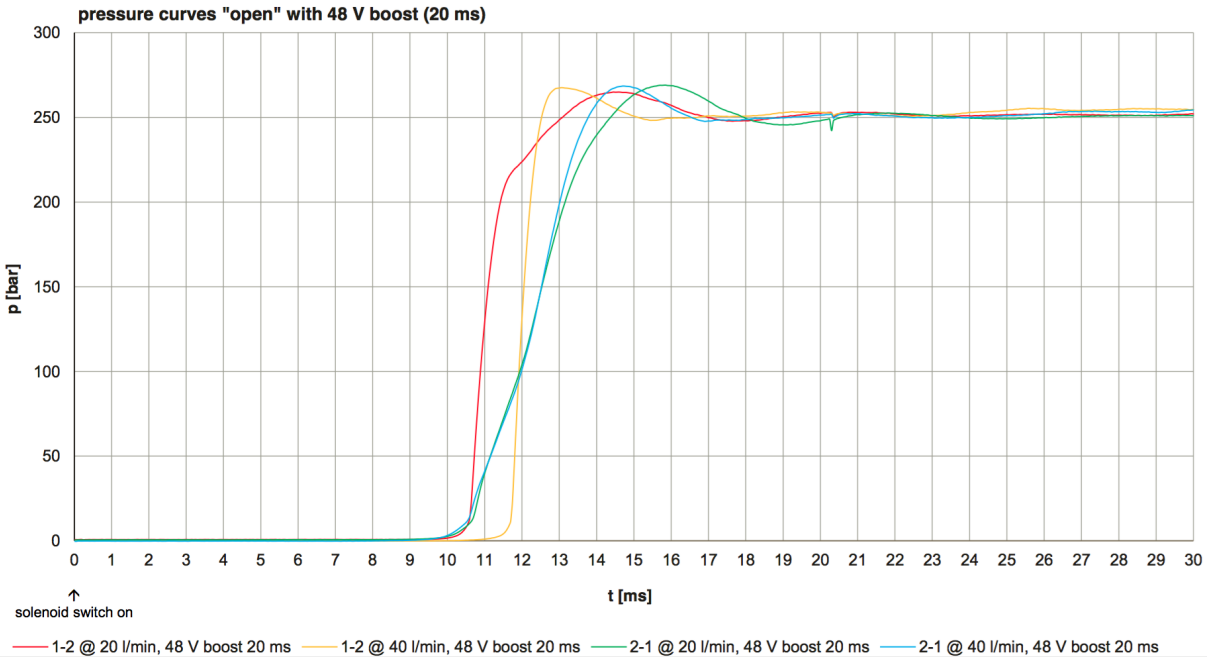


Figure 5.3. Input to pressure time response for the manifold valves. 1-2 and 2-1 refers to flow direction.[21]

5.1.2 Transmission Line Dynamics

With the fast switching valves, pressure transients in the transmission lines connecting the cylinder chambers to the manifold are expected. In order to take the pressure transients into account a transmission line model is made. The transmission line model is made as a lumped parameter model, where the transmission line is divided into a number of elements. For each element pressure, flow and resisting force are calculated. The transmission line model is illustrated in figure 5.4.

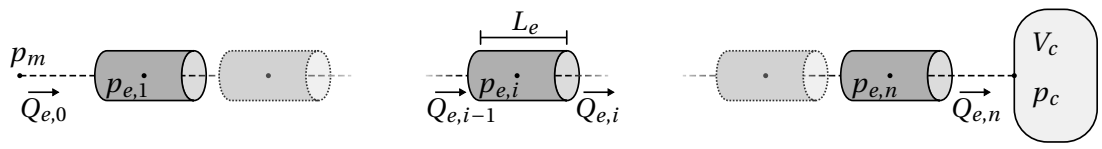


Figure 5.4. Illustration of elements in transmission line model.

As illustrated in the figure the transmission line is divided into a number of elements. $p_{e,i}$ denotes the pressure of the i'th element. The flow $Q_{e,i}$ is defined as the flow from element i into $i+1$. $Q_{e,0}$ is the flow into the transmission line and $Q_{e,n}$ is the flow out of the transmission line. For the transmission line from the manifold to the chamber it follows that; $Q_{e,0} = Q_m$ and $Q_{e,n} = Q_c$.
The momentum and continuity equation are applied to each element. Applying the momentum and continuity equation to the i'th element yields [22]

$$\dot{Q}_{e,i} = \frac{(p_{e,i} - p_{e,i+1})A_{e,i} - \Delta p_{fric,i} A_{e,i}}{\rho L_{e,i}} \tag{5.9}$$

$$\dot{p}_{e,i} = \frac{(Q_{e,i-1} - Q_{e,i})\beta_e(p_{e,i})}{A_{e,i} L_{e,i}} \tag{5.10}$$

where A_e is the cross sectional area of the element, L_e is the length of the element, ρ is the density of the fluid and Δp_{fric} is the pressure loss due friction and fittings connecting the hose/pipe. For the i'th element with n fittings the total pressure drop may be written as

$$\Delta p_{fric,i} = \Delta p_{\lambda,i} + \Delta p_{\xi,1,i} + \Delta p_{\xi,2,i} + \cdots + \Delta p_{\xi,n,i} \tag{5.11}$$

where Δp_λ and Δp_ξ are the pressure loss due to friction and fittings respectively. The pressure loss due to friction as the fluid flow through the pipe, can be calculated for the i'th element using Darcy's equation. [23]

$$\Delta p_{\lambda,i} = \lambda_i \frac{L_{e,i}}{d_{e,i}} \rho \frac{1}{2} \left(\frac{Q_{e,i}}{\frac{1}{4} d_{e,i}^2 \pi} \right)^2 \tag{5.12}$$

where d_e is the diameter of the element and λ is the friction factor, which can be expressed as function of Reynold's number

$$\lambda = \frac{64}{Re} \qquad \text{For laminar flow (Re<2300)} \tag{5.13}$$

$$\lambda = \frac{0.3164}{Re^{0.25}} \qquad \text{For laminar flow (Re>2300)} \tag{5.14}$$

Reynold's number is calculated as

$$\mathrm{Re} = \frac{\rho v_{e,i} d_{e,i}}{\mu} \tag{5.15}$$

where v is the mean velocity of the fluid and μ is the dynamic viscosity.

In order to increase the robustness of the simulation and not have discontinuity at Re=2300 a hyperbolic-tangent expression is used to create a smooth transition between laminar and turbulent flow. With the hyperbolic-tangent expression the pressure loss due to friction is modelled as

$$\Delta p_{\lambda,i} = \frac{64}{Re} \frac{L_{e,i}}{d_{e,i}} \rho \frac{1}{2} \left(\frac{Q_{e,i}}{\frac{1}{4} d_{e,i}^2 \pi} \right)^2 \left(\frac{1}{2} + \frac{\tanh\left(\frac{2300 - Re}{50}\right)}{2} \right) + \frac{0.3164}{Re^{0.25}} \frac{L_{e,i}}{d_{e,i}} \rho \frac{1}{2} \left(\frac{Q_{e,i}}{\frac{1}{4} d_{e,i}^2 \pi} \right)^2 \left(\frac{1}{2} + \frac{\tanh\left(\frac{-2300 + Re}{50}\right)}{2} \right) \tag{5.16}$$

The pressure loss due to fitting n for the i'th element is modelled as [23]

$$\Delta p_{\xi,n,i} = \frac{\xi_{n,i} \rho}{2} \left(\frac{Q_{e,i}}{\frac{1}{4} d_{e,i}^2 \pi} \right)^2 \tag{5.17}$$

where ξ is a friction coefficient for the given fitting type.

5.2 Modelling of Load Side Cylinder

In this section the equations used to model the load side cylinder will be presented. The load side consist of a symmetric hydraulic cylinder connected to two proportional valves in parallel. The load side is supplied by two hydraulic pumps of 125cc and 250cc. The pumps are pressure controlled, so the pump pressure can be seen as a constant pressure, p_P, up to 350bar.

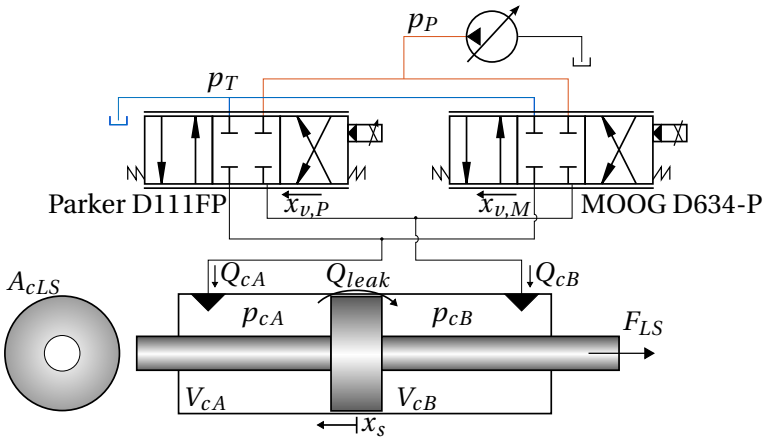


Figure 5.5. Hydraulic schematic of load side cylinder

5.2.1 Pressure Build-up in Cylinder Chambers

The transient response of the cylinder pressures are modelled using the continuity equation. These are expressed as

$$Q_{cA} - Q_{leak} = -\dot{x}_s A_{cLS} + \frac{V_{cA}(x_s)}{\beta_e(p_{cA})}\dot{p}_{cA} \tag{5.18}$$

$$Q_{cB} + Q_{leak} = \dot{x}_s A_{cLS} + \frac{V_{cB}(x_s)}{\beta_e(p_{cB})}\dot{p}_{cB} \tag{5.19}$$

The leakage flow Q_{leak} is as seen in equation 5.20.

$$Q_{leak} = c_{leak}(p_{cB} - p_{cA}) \tag{5.20}$$

Where c_{leak} is the leakage coefficient. The chamber volumes as function of cylinder position are calculated as

$$V_{cA}(x_s) = \left(\frac{L_{LS}}{2} - x_s\right) A_{cLS} + V_{hose} \tag{5.21}$$

$$V_{cB}(x_s) = \left(\frac{L_{LS}}{2} + x_s\right) A_{cLS} + V_{hose} \tag{5.22}$$

Where L_{LS} is the total stroke length of the cylinder. V_{hose} is the hose volume which is equal for both chambers.

5.2.2 Valve Characteristics

As mentioned the cylinder is controlled by two proportional valves in parallel. Because of the the dead band around zero flow velocity in the larger 1000L/min Parker D111FP valve, the smaller 100L/min MOOG D634 valve is introduced to compensate for this. The valves are first modelled individually.

The nominal spool position is defined from -1 to 1, therefore each flow out of the valves is expressed for a positive and negative spool reference. The flow through the valves are calculated using the orifice equation. For the Parker valve the orifice equation is expressed as in equations 5.23 and 5.24.

$$Q_{cA,P} = \begin{cases} c_d A_d(x_{v,P}) \sqrt{\frac{2}{\rho}|p_P - p_{cA}|}\text{sgn}(p_P - p_{cA}) & , \quad x_{v,P} \geq 0 \\ -c_d A_d(x_{v,P}) \sqrt{\frac{2}{\rho}|p_{cA} - p_T|}\text{sgn}(p_{cA} - p_T) & , \quad x_{v,P} < 0 \end{cases} \tag{5.23}$$

$$Q_{cB,P} = \begin{cases} -c_d A_d(x_{v,P}) \sqrt{\frac{2}{\rho}|p_{cB} - p_T|}\text{sgn}(p_{cB} - p_T) & , \quad x_{v,P} \geq 0 \\ c_d A_d(x_{v,P}) \sqrt{\frac{2}{\rho}|p_P - p_{cB}|}\text{sgn}(p_P - p_{cB}) & , \quad x_{v,P} < 0 \end{cases} \tag{5.24}$$

Where $A_d(x_{v,P})$ is the opening area of the valve as a function of spool position $x_{v,P}$, and ρ is the density of the fluid.

For the MOOG valve the orifice equations can be written as equations 5.25 and 5.26. The valve coefficient $k_{v,M}$ is found from the datasheet with $k_{v,M} = \frac{Q_{nom}}{\sqrt{\Delta p_{nom}}}$.

$$Q_{cA,M} = \begin{cases} k_{v,M} x_{v,M} \sqrt{|p_P - p_{cA}|}\text{sgn}(p_P - p_{cA}) & , \quad x_{v,M} \geq 0 \\ k_{v,M} x_{v,M} \sqrt{|p_{cA} - p_T|}\text{sgn}(p_{cA} - p_T) & , \quad x_{v,M} < 0 \end{cases} \tag{5.25}$$

$$Q_{cB,M} = \begin{cases} -k_{v,M} x_{v,M} \sqrt{|p_{cB} - p_T|}\text{sgn}(p_{cB} - p_T) & , \quad x_{v,M} \geq 0 \\ -k_{v,M} x_{v,M} \sqrt{|p_P - p_{cB}|}\text{sgn}(p_P - p_{cB}) & , \quad x_{v,M} < 0 \end{cases} \tag{5.26}$$

So; $Q_{cA} = Q_{cA,M} + Q_{cA,P}$ and $Q_{cB} = Q_{cB,M} + Q_{cB,P}$.

Valve Dynamics

The position dynamics for both valves are described using a linear second order model with parameters taken from the respective data sheet. The natural eigenfrequency are 54 and 30 Hz, and the damping factor is 0.52 and 0.707 for the MOOG and Parker valve respectively. A bode plot of the two valves are shown in figure 5.6 and datasheet bode plots are show in figure 5.7 and 5.8.

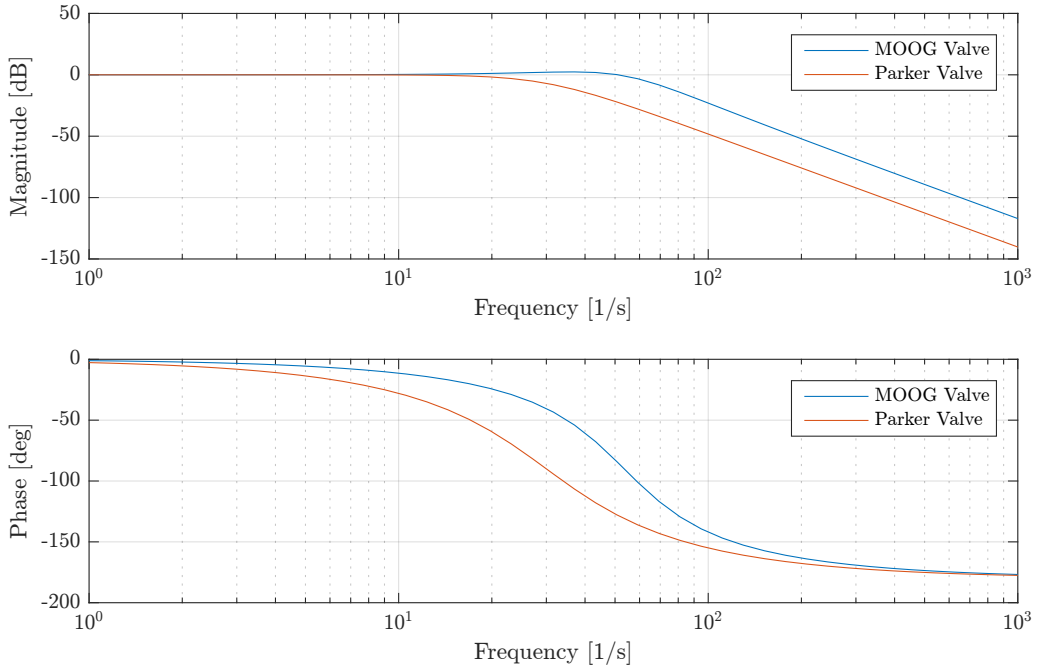


Figure 5.6. Bode plot of MOOG and Parker valve dynamics.

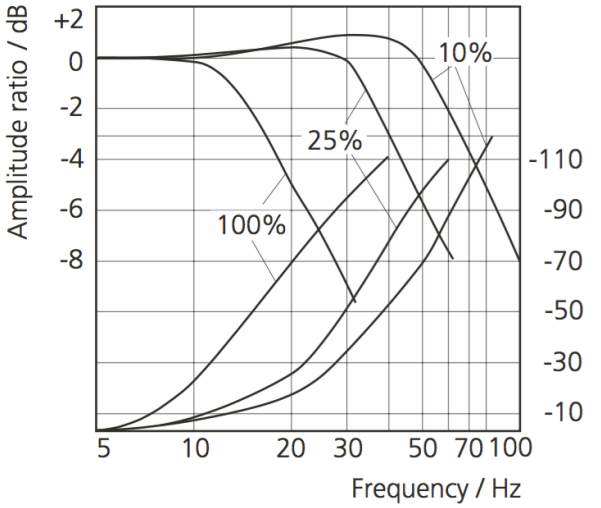


Figure 5.7. MOOG valve data sheet bode plot. 10% value is used.

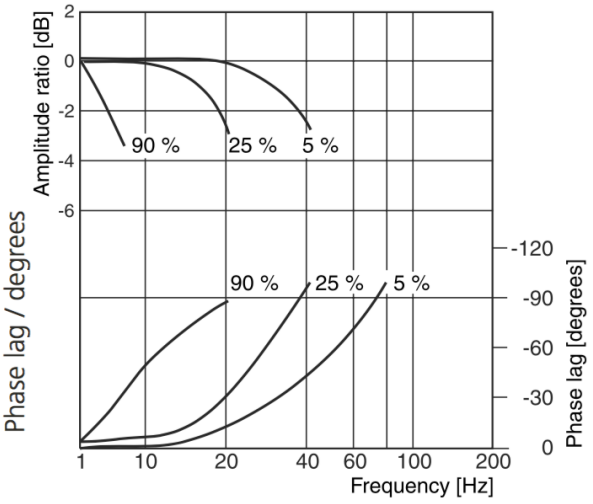


Figure 5.8. Parker valve data sheet bode plot. 5% value is used.

Signal Splitting between the Valves

From the datasheets of valves a characteristic of the valve opening can be calculated. The valve characteristic as function of nominal spool position are shown on figure 5.9. Here it can be seen that there is a clear non-linearity of the Parker valve with the dead band around zero spool position as mentioned earlier. Also the linear relation of the MOOG-valve is illustrated on the figure.

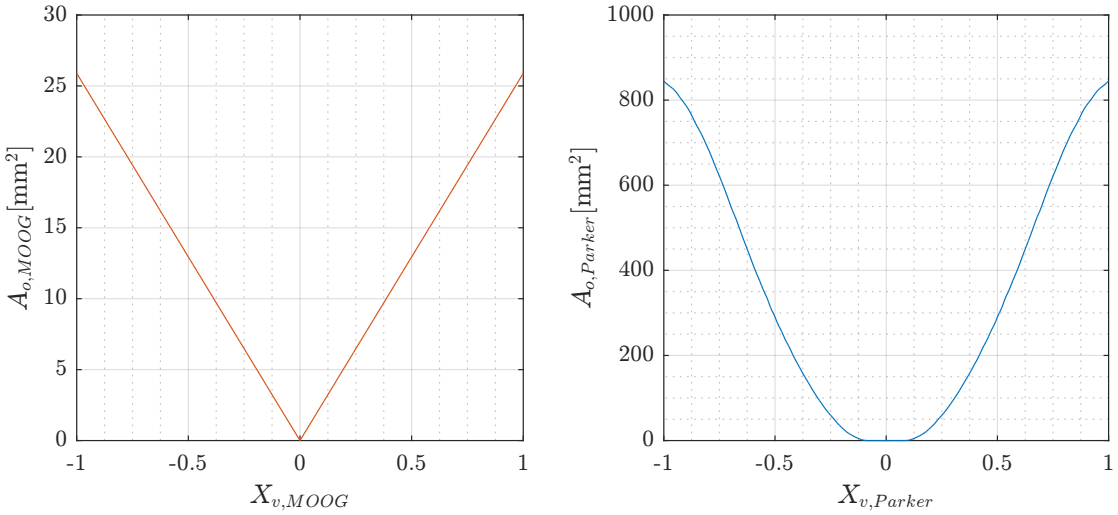


Figure 5.9. Opening area of MOOG- and Parker-valve as function of nominal spool position.

The signal to each of the valves is then calculated using a "splitting algorithm". In order to have the best possible performance of the load side cylinder two different algorithms are investigated. The algorithms are elaborated in appendix G on page 171.

5.3 Mechanical System

The mechanical system consists of two cylinder pistons and the sleigh connecting them, and thus the mass considered is the total moving mass of this system. The forces working on the sleigh are the cylinder force from each cylinder and mechanical friction. The system is regarded as a stiff system, e.g bending moments resulting from sleigh misalignment are not considered. The dynamics of this system are then described with a force equilibrium based on figure 5.10.

$$m_s\ddot{x}_s = F_{DDC} - F_{LS} - F_{fric}(\dot{x}_s) \tag{5.27}$$

where m_s is the total sleigh mass including piston masses. Furthermore

$$F_{DDC} = -A_{c1}p_{c1} + A_{c2}p_{c2} + A_{c3}p_{c3} + A_{c4}p_{c4} \tag{5.28}$$

$$F_{LS} = A_{cLS}(p_{cA} - p_{cB}) \tag{5.29}$$

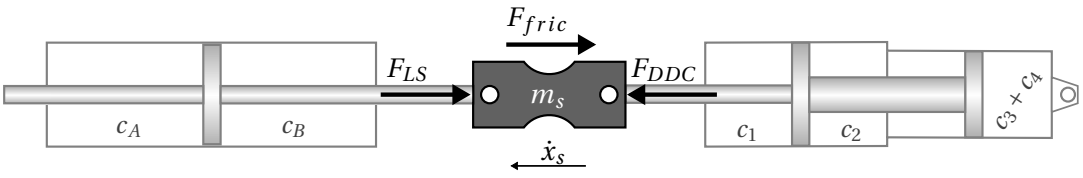


Figure 5.10. Force equilibrium.

F_{fric} is a sum of the Coulomb-friction and viscous friction. For ease of simulation, the Coulomb friction is modelled with a tangent hyperbolic function to smooth out the function around $\dot{x}_s = 0$.

Hence, the friction term is described by

$$F_{fric} = \tanh\left(\frac{\dot{x}_s}{\gamma}\right)\left(F_{C1} + F_{C2}e^{-|\dot{x}_s|/k_{stri}}\right) + B_{vs}\dot{x}_s \tag{5.30}$$

where γ is a coefficient used to control the slope of the tangent hyperbolic function around $\dot{x}_s = 0$, F_{C1} is the Coulomb friction constant and F_{C2} and k_{stri} are coefficients related to the stiction-type Stribeck friction. B_{vs} is the viscous friction coefficient. All friction coefficients are assumed constant and are to be determined experimentally. In appendix A.1 on page 125 the friction is determined.

5.4 Model Overview

The equations described in this chapter are combined to simulate the entire system. A graphical illustration of the model is seen in figure 5.11. The model have been implemented in MATLAB Simulink.

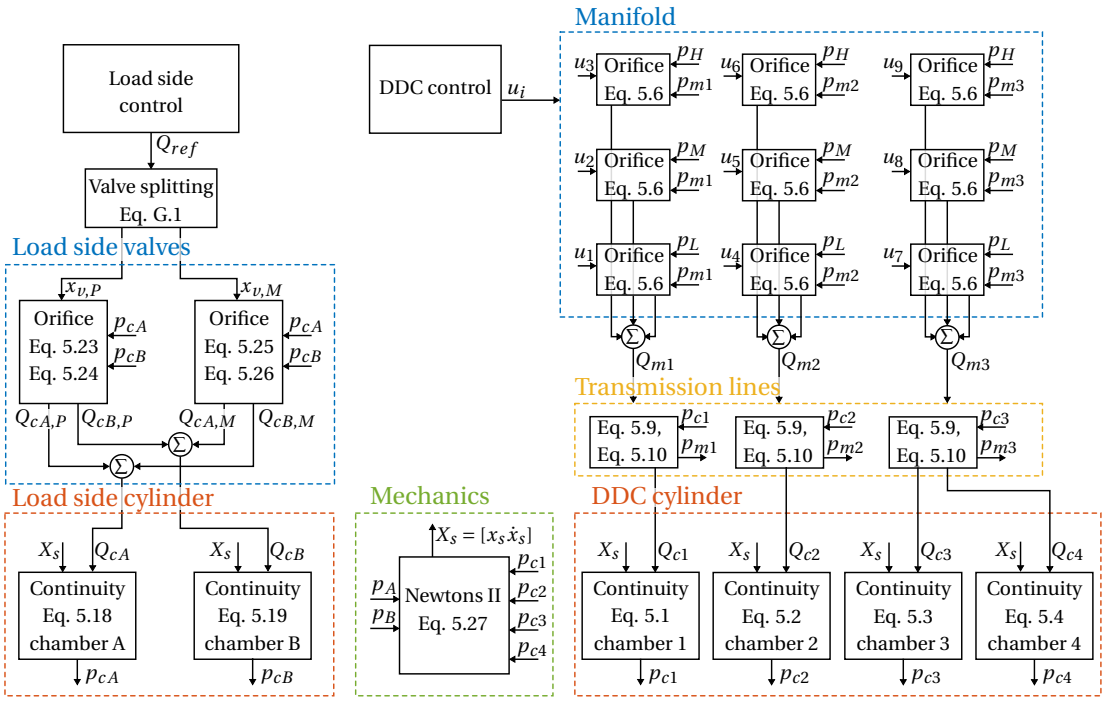


Figure 5.11. Overview of system model.

5.5 Energy Losses

In this section the energy losses of the DDC are investigated. A range of different losses are present in a hydraulic cylinder. The focus in this section are mainly on the losses associated with the valve-manifold in form of the compressibility losses associated with shifts between pressure levels and losses due to throttling of the valves. In [12] a study on minimising the losses in a DDC are investigated. Based on this research the losses in this section are evaluated.

5.5.1 Switching losses

The first losses evaluated are compression losses caused by a switching between different fixed pressures for a certain volume. In equation 5.31 the power loss from connecting a volume with

pressure, p_{old}, to a fixed pressure supply with pressure, p_{new}. This power loss is defined as the difference between energy supplied by the fixed pressure line, E_S, and the potential pressure energy save in the volume E_V.

$$E_{\beta,loss} = E_S - E_V = \frac{1}{2}(p_{new} - p_{old})^2 \frac{V}{\beta} \tag{5.31}$$

where V is chamber volume and β is effective bulk modulus.

The total switching loss for the entire cylinder is then calculated as the sum of losses in all three chambers

$$E_\beta = \sum_{i=1}^{3} \frac{1}{2} \left(p_{new,i} - p_{old,i}\right)^2 \frac{V_i(x(t))}{\beta} \tag{5.32}$$

where the subscript i denotes the chambers. The chamber volumes are seen as equation 5.33.

$$\begin{aligned} V_1(\mathbf{x}(t)) &= (L_s - x_s) A_{c1} + V_{0c1} \\ V_2(\mathbf{x}(t)) &= x_s A_{c2} + V_{0c2} \\ V_3(\mathbf{x}(t)) &= x_s (A_{c3} + A_{c4}) + V_{0c3} + V_{0c4} \end{aligned} \tag{5.33}$$

where A_c is cross-sectional area of chamber, V_{0c} is the initial volume of the given chamber, and L_s is the stroke length. Since chamber 3 and 4 are connected in parallel these are seen together.

The design of the manifold gives, as mentioned, 27 different configurations of pressure levels in the chambers. In figure 5.12, 5.13 and 5.14 switching between the 27 configurations are illustrated, for three different cylinder positions. The figures shows a clear symmetry around the diagonal from (0,0) because the losses due to shifting are the same for e.g. high to low as low to high.

An interesting observation about the switching loss, is that the loss from shifting from low to high pressure in one chamber is bigger than the sum of shifting from low to medium and medium to high in the same chamber, and vice versa for high to low. The phenomena is expressed in equation 5.34.

$$\begin{aligned} &\frac{1}{2}(p_H - p_L)^2 \frac{V}{\beta} > \left(\frac{1}{2}(p_M - p_L)^2 \frac{V}{\beta} + \frac{1}{2}(p_H - p_M)^2 \frac{V}{\beta} \right) \\ \implies &(p_H - p_L)^2 > \left((p_M - p_L)^2 + (p_H - p_M)^2 \right) \\ \implies &p_H^2 + p_L^2 - 2p_H p_L > p_M^2 + p_L^2 - 2p_M p_L + p_H^2 + p_M^2 - 2p_H p_M \\ \implies &- p_H p_L > p_M^2 - p_M p_L - p_H p_M \\ \implies &0 > p_L(p_H - p_M) + p_M(p_M - p_H) \\ \implies &0 > p_L(p_H - p_M) - p_M(p_H - p_M) \end{aligned} \tag{5.34}$$

Knowing that $p_H > p_M$, it follows that $(p_H - p_H) > 0$. And since $p_M > p_L$, the inequality above is strictly true.
From this it becomes evident that is possible to decrease the cost of switching by going through an intermediate pressure level. Hence, it would be possible to decrease the overall switching losses further by increasing the amount of intermediate pressure levels in the system.

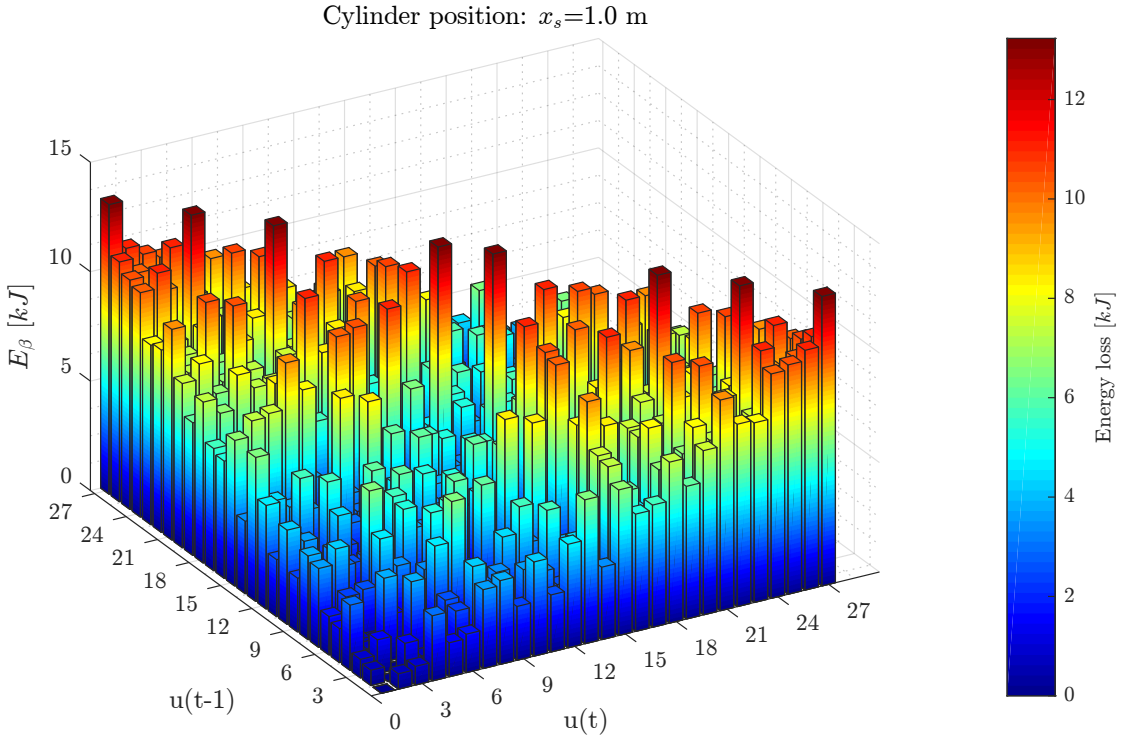


Figure 5.12. Losses from switching between the 27 force levels for $x_s = 1m$.

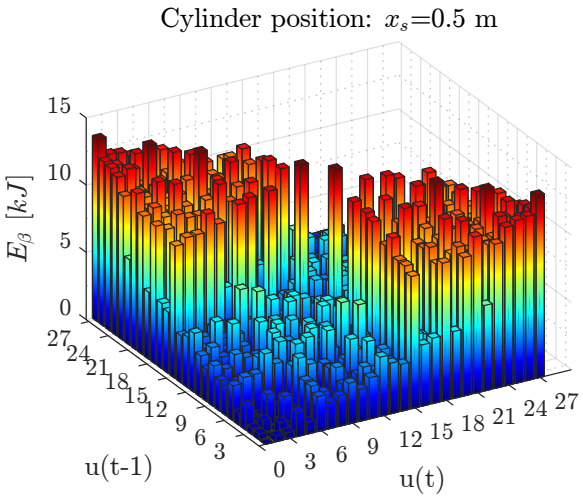


Figure 5.13. Losses from switching between the 27 force levels for $x_s = 0.5m$.

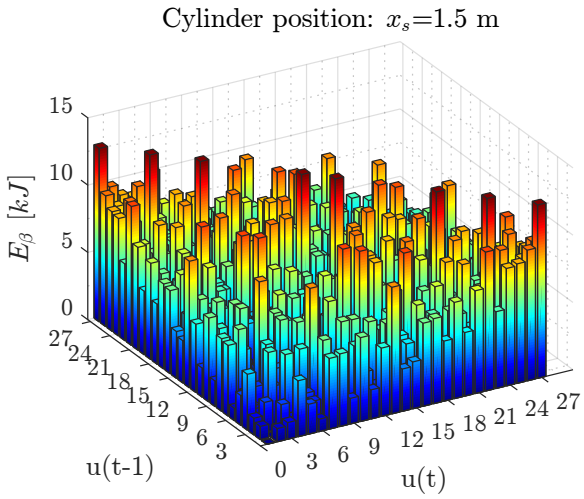


Figure 5.14. Losses from switching between the 27 force levels for $x_s = 1.5m$.

5.5.2 Throttling Losses

The other loss evaluated is the throttling losses. Each chamber is always connected to one pressure line through the valve-manifold. The throttling loss is associated with the loss through each of these valves [12]. The throttling loss is derived from the orifice equation in section 5.1.1 on page 20 and is

expressed as

$$E_{T,loss} = |\Delta p_v Q_v| = \frac{Q_v^2}{(n_i k_v)^2}|Q_v| = \frac{|\dot{x}_s^3 A_c^3|}{(n_i k_v)^2} \tag{5.35}$$

where Δp_v is pressure drop across the valve, Q_v is flow over the valve, k_v is the valve gain and n_i is the number of valves for the specific chamber.

The total loss caused by throttling is then the sum over all active valves. This can be written as:

$$E_T = |\dot{x}_s^3| \sum_{i=1}^{3} \frac{A_{c,i}^3}{(n_i k_v)^2} \tag{5.36}$$

where the subscript i denotes the different chambers. Again due to the parallel connection of chamber 3 and 4, these chambers are evaluated together. This means that for $i = 3$ the cross-sectional area is $A_{c3} + A_{c4}$.

From the equation it is seen that the throttling losses are dependent on the sleigh velocity cubed, but it is independent of the chamber pressures. In figure 5.15 the throttling loss as a function of sleigh velocity is shown. It can be seen that the throttling loss is insignificant for low velocities.

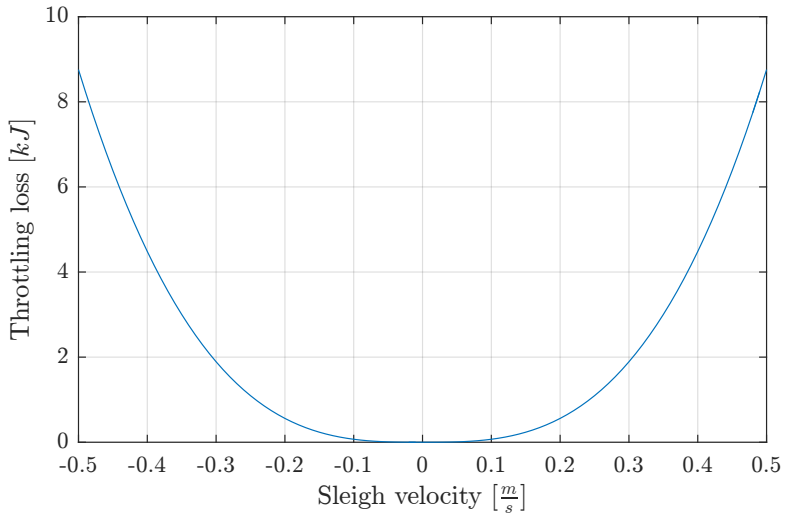


Figure 5.15. Throttling loss as function of sleigh velocity.

5.6 Chapter Summary

In this chapter the model of the DDC, load side cylinder and the mechanical system has been presented. The model of the DDC included a description of the cylinder itself, the manifold containing nine DFCUs and the transmission line dynamics, which has been included, due to the length of the transmission lines between the manifold and the cylinder chambers. The symmetric hydraulic load side cylinder has been described, together with a description of the two proportional valves in parallel. The signal to the valves is divided between the two valves by the use of a "splitting algorithm", which is elaborated in appendix G. The mechanical is described by a force equilibrium, leading to a combined model of the entire system. Lastly, the energy losses associated with switching between the discrete force levels and the throttling losses of the manifold are presented.

In the following chapter the derived model of the system, will be validated by comparing the simulated model with experimental data for the test setup.

CHAPTER 6

VALIDATION OF NON-LINEAR MODEL

To validate the parameters of the non-linear model, some simple experiments have been conducted on the test setup. The experimental results are then compared, with the simulation of mathematical model and parameters are adjusted, to make the model match the experimental results.

6.1 Load Side Validation

First the load side model is isolated to validate the orifice equations and continuity equations of the load side, and the mechanics of the whole system.

The most important soft parameters that are evaluated are bulk-modulus, the valve gains of the two proportional valves and the friction model. Bulk-modulus mainly affect the frequency of pressure and velocity dynamics. The valve gains primarily effect the steady state value of the velocity. The amplitude of the dynamics and the static load pressure value is mainly affected by the friction model. All parameters have different impact on the system and final values presented in table 6.1 on page 39 are found iteratively.

The valve flow gains are validated by performing a number of test, with different spool reference inputs to each valve individually. The valve gains are then adjusted to match the simulated velocity with the measured velocity. For the MOOG valve the valve gain, $k_{v,M}$, is linear and can therefore be evaluated directly. This is visualised in figure 6.1, where the left subplot shows the velocity at steady state for inputs in a range of [-5:5]V. The right subplot shows the valve gain as function of nominal input. The valve gain is determined from the orifice equation by multiplying the velocity with the cylinder area divided by the pressure difference squared at the given velocity.

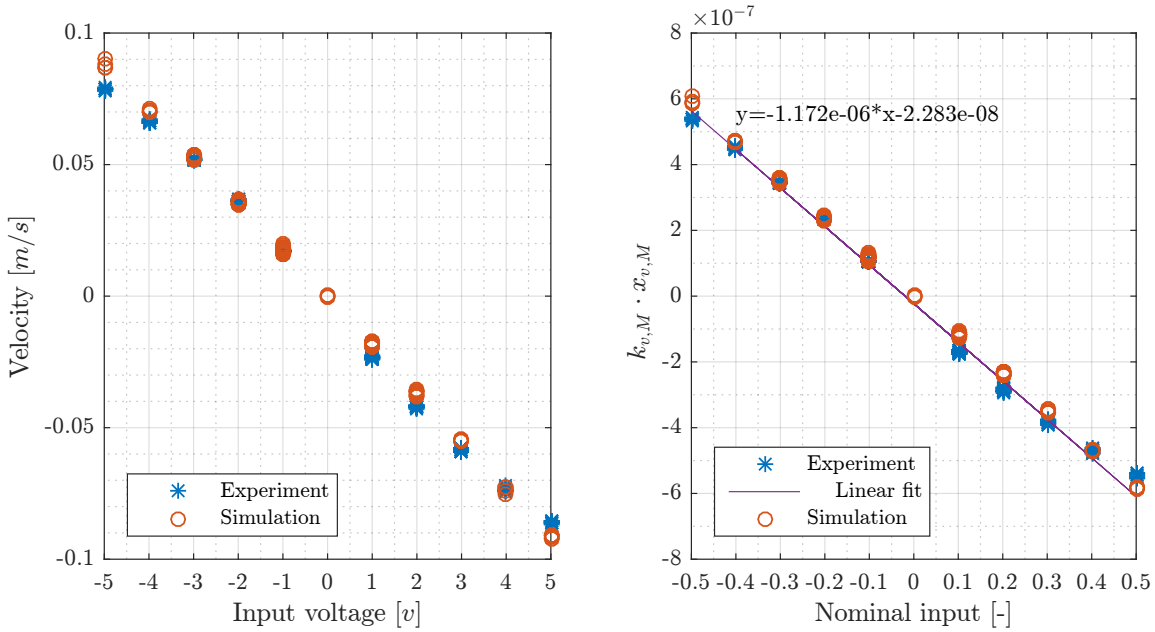


Figure 6.1. To the left; input to velocity response of MOOG valve. To the right; valve gain of MOOG valve with a linear fit to find $k_{v,M}$.

For the larger Parker valve the input to velocity is validated by evaluating the non-linear opening characteristics given in the datasheet. Due to the size of the Parker valve, it is only possible to evaluate the opening in a relatively small input range; [-2:2]V. The results are seen in figure 6.2. The figure shows that the modelled opening characteristics based on the datasheet, is increased to match the experiment.

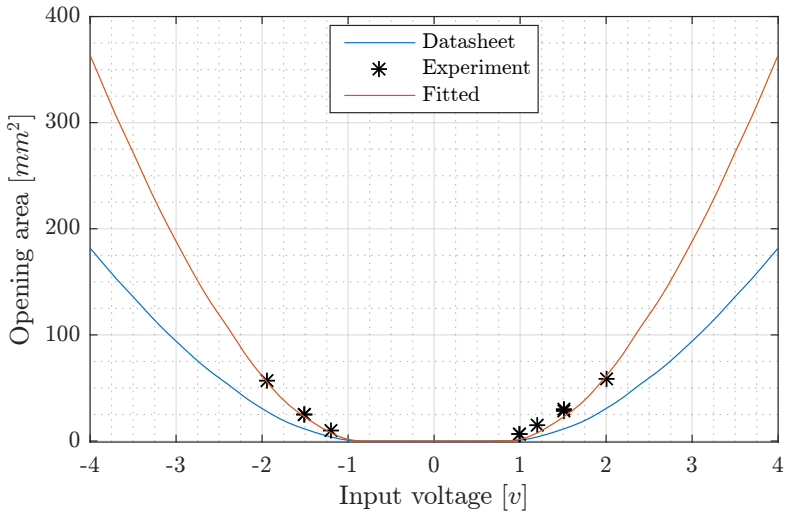


Figure 6.2. Opening characteristics of Parker valve from datasheet, experiment and the fitted curved used in the model.

Implementation of the measured friction model, seen in appendix A.1, gave some problems when simulating the system due to a high number of zero crossings. This was solved by reducing the static friction curve by removing some stiction and increasing viscous friction. This compromise cause a

smaller steady state error in the modelled load pressure, when compared to the real system. For the scope of this thesis it is acceptable, as long as the system dynamics have similar response.

To make the velocity and load pressure dynamics of the model match the experiment, the value for bulk-modulus have been evaluated together with the friction coefficients. In figure 6.3 the velocity response and load pressure during a step from +4v to -4v on the MOOG valve is seen. The simulated and measured results are similar. Though the amplitude of the simulated velocity dynamics are larger, the settling time and frequency matches the experiment.

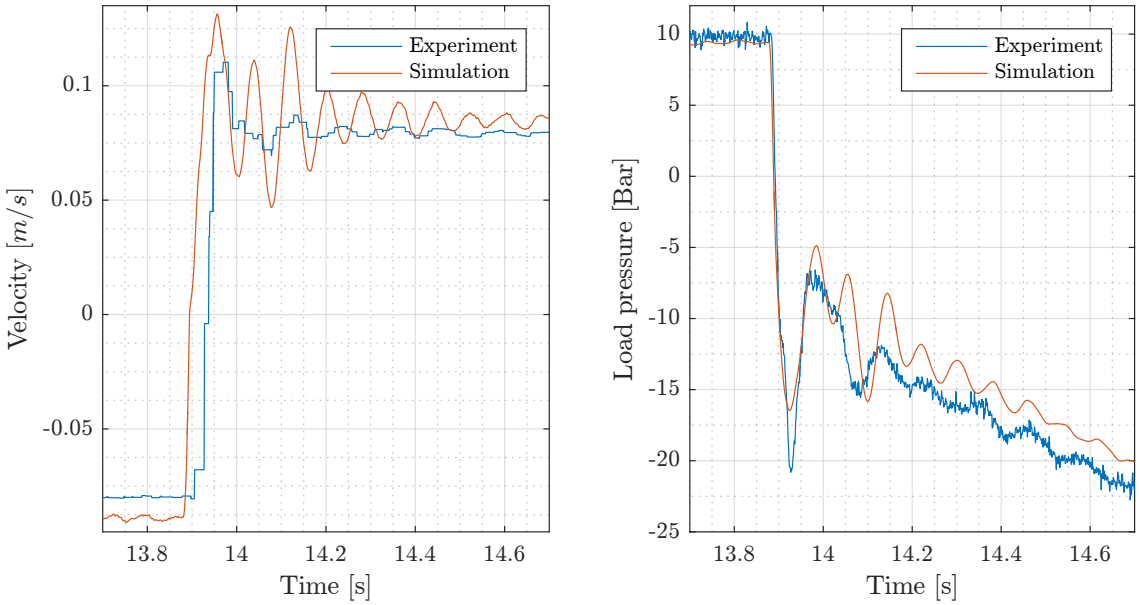


Figure 6.3. Step from +4v to -4v on the MOOG valve. To the left velocity, to the right load pressure.

6.2 DDC-side Validation

To validate the pressure dynamics of the DDC-side an experiment where the load side is set to hold a constant position is conducted. A range of different force steps have been applied to the DDC-manifold. The Simulink model is then applied with the same valve inputs and secondary side pressures as measured during the experiment. With this approach it is possible to validate the parameters of the continuity and orifice equations of the DDC-side. Also the line dynamics from manifold to cylinder chambers are validated. It is assumed that the difference in pressure for chamber 3 and 4 is insignificant, thus these chambers are seen as one chamber with the same pressure and with summed piston area and chamber volume. This assumption is therefore initially validated.

6.2.1 Parallel Connection of Chamber 3 and Chamber 4

To show that the chambers 3 and 4 can be seen as one chamber the response from different shifts are investigated. In figure 6.4 the measured pressure for chamber 3 and 4 is shown together with the response from a simulation where the two chambers modelled as one, by adding chamber volumes and piston areas in the continuity equation. The figure shows similar dynamic pressure responses in the measured data from both chambers with an insignificant difference in amplitude and frequency of the pressure oscillations. The simulated response also shows similar characteristics regarding the

pressure dynamics. Thus, for the rest of this thesis the parallel connected chamber 3 and chamber 4 will be noted together as chamber 3'. This leads to

$$c3' = c3 + c4 \tag{6.1}$$

$$p_{c3'} \approx p_{c3} \approx p_{c4} \tag{6.2}$$

$$Q_{c3'} = Q_{c3} + Q_{c4} \tag{6.3}$$

$$A_{c3'} = A_{c3} + A_{c4} \tag{6.4}$$

$$V_{0,c3'} = V_{0,c3} + V_{0,c4} \tag{6.5}$$

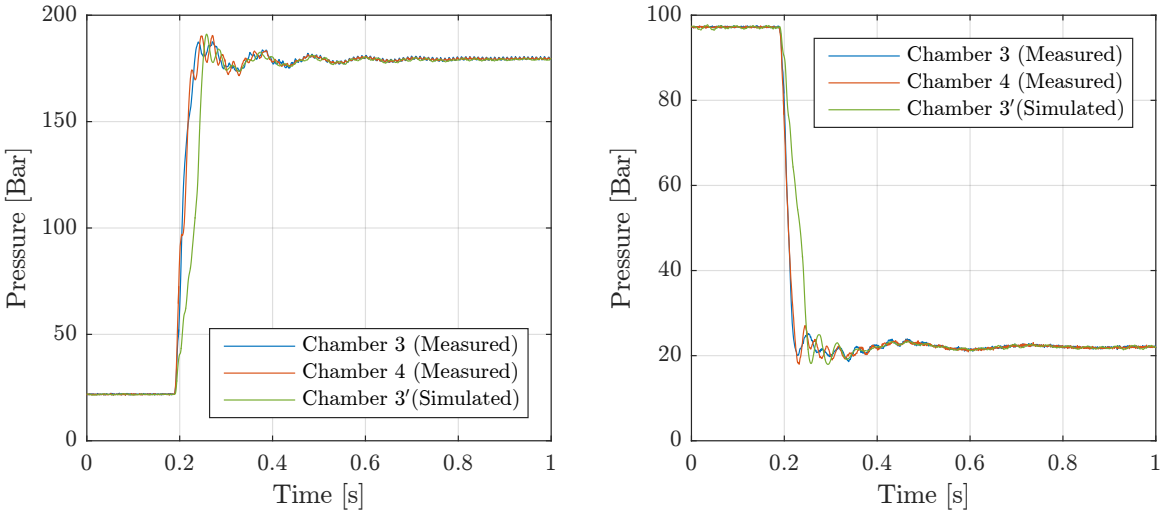


Figure 6.4. Measured pressures for chamber 3 and 4 and simulated pressure for chamber 3', where the two chambers modelled as one. Left is step from low to high pressure. Right is step from medium to low pressure.

6.2.2 Pressure Dynamics

All chamber pressures are shown for a small force shift from force level 9 to 8 on figure 6.5 and a larger step from force level 1 to 14 is seen on figure 6.6. Good similarity between measured and simulated response for most cases is seen. However, the validation showed a small difference between measurements and simulations for small force step during the first 50 ms. This can be caused by the opening characteristics of the Bucher on/off valves which are dependent of both flow and flow direction [21]. Generally the pressure dynamics of the model looks similar to the measured. Thus, the continuity and orifice equations of the DDC-side are validated.

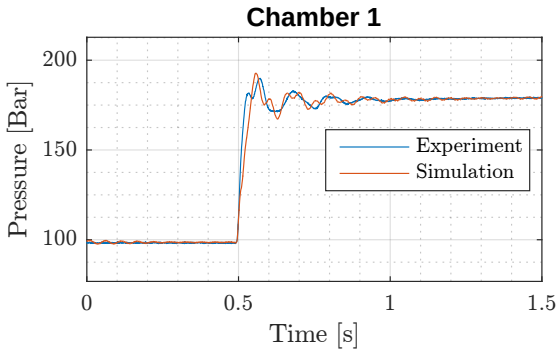


Figure 6.5. Chamber pressures for a step from force level 9 to 8.

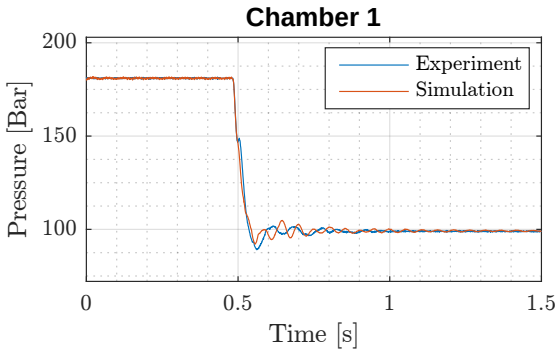
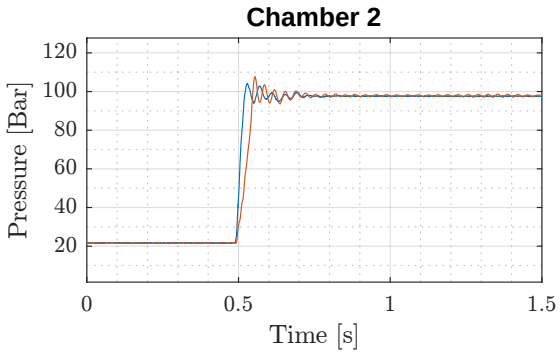
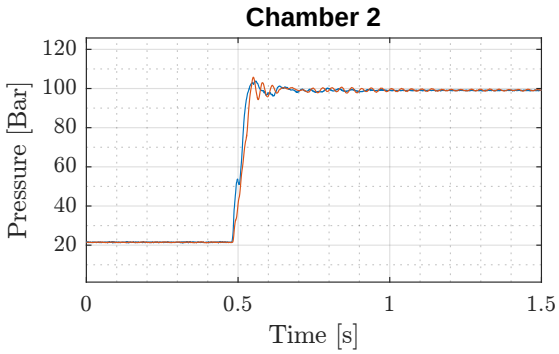
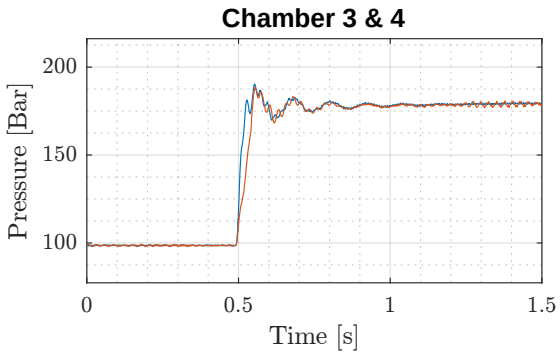
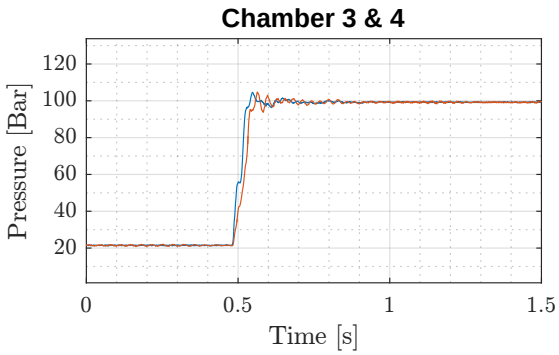


Figure 6.6. Chamber pressures for a step from force level 1 to 14.









6.2.3 Pressure Dynamics in Transmission Lines

Next the pressure dynamics introduced by the transmission line (pipes, hoses and fittings) connecting the cylinder to the manifold are validated. In figure 6.7 the pressure response is shown at the chamber and at the manifold, before the transmission line, for a force shift from level 9 to 8. The figure clearly shows how the transmission line dynamics introduces extra pressure oscillation in the cylinder chambers. Furthermore it is seen that the model pressure dynamics matches the experiment fairly well at both ends of the transmission line.

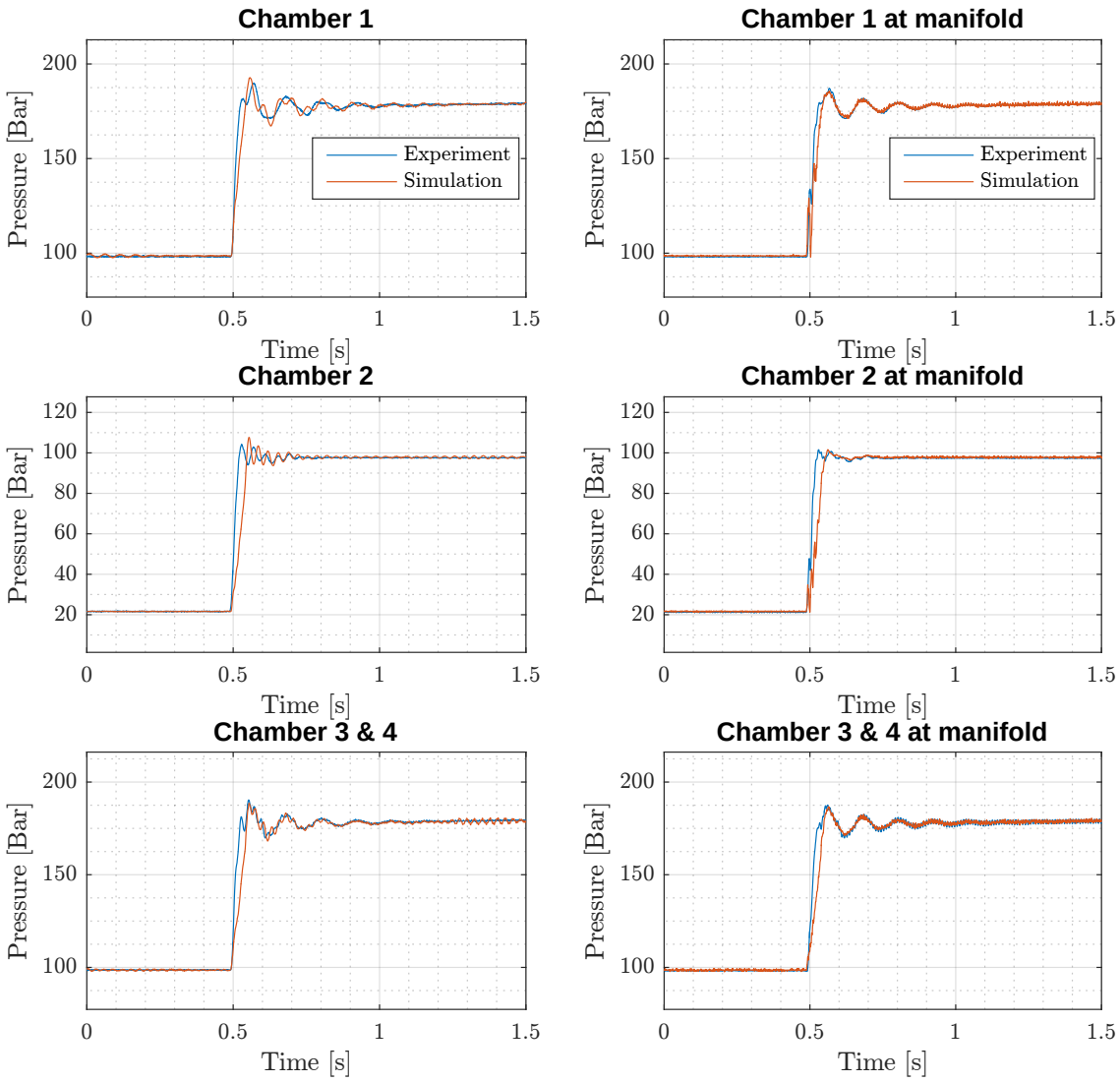


Figure 6.7. Pressure response for each chamber before and after the transmission lines connecting the chambers to the manifold. Left column is pressures in the chambers, right column is pressure at the manifold.

6.3 Chapter Summary

In this chapter the non-linear model has been validated by comparison with experimental data from the test setup. For the load side, the valve gains was fitted and validate based on experiments with different constant velocities. Implementation of the friction model determined based on measured data, led to simulation issues and resultantly the friction model was adjusted. Even though a compromise was made on the friction curve, a joint evaluation of bulk-modulus and friction parameters showed similarity between measurements and simulations for velocity and load pressure responses. Evaluation of the DDC side showed that it was possible to make the assumption, that the two parallel connected chambers can be seen together as one. The DDC validation further showed close similarities in the chambers pressure dynamics, as well as the transmission line dynamics. Subsequently the model was considered sufficiently accurate for the following control development.

In table 6.1 the soft parameters found in the validation are listed.

Hydraulic properties		
- Bulk modulus	β_{max}	6000 [Bar]
- Dynamic viscosity of oil	μ	0.0810 [Pa·s]
- Density of fluid	ρ	860 [kg/m^3]
Friction properties		
- Coulomb friction	F_{C1}[N]	6000 [N]
- Stribeck friction	F_{C2}[N]	6000 [N]
- Stribeck slope coefficient	k_{stri} [m/s]	0.2
- Friction slope coefficient	γ	1e-3
- Viscous friction	B_{vs} [Ns/m]	60000 [Ns/m]
MOOG Valve (Load side)		
- Valve gain	$k_{v,M}$	1.172e-06 [m^3/(s$\sqrt{\text{Pa}}$)]
- Natural eigenfrequency	$\omega_{nv,M}$	53.5 [Hz]
- Damping factor	$\zeta_{v,M}$	0.52
Parker Valve (Load side)		
- Valve gain / Opening characteristics	[-]	*See figure 6.2*
- Natural eigenfrequency	$\omega_{nv,P}$	30 [Hz]
- Damping factor	$\zeta_{v,P}$	0.707
Bucher on/off Valve (DDC side)		
- Valve gain	k_v	9.8995e-07 [m^3/(s$\sqrt{\text{Pa}}$)]
- Natural eigenfrequency	ω_{nv}	100 [Hz]
- Damping factor	ζ_v	0.707

Table 6.1.

Part II

Control Design

CHAPTER 7

CONTROL CONSIDERATIONS

This chapter discusses some general control consideration regarding the system. The discussion is made on the basis of the system to be controlled and thus general for the system regardless of the type of control being implemented. Furthermore, an application case to be used for control evaluation is presented.

7.1 System Analysis

In this section a commencing analysis of the control essential elements of the system described in chapter 3 is discussed. This analysis is then used as a basis for the control design in the following chapters.

7.1.1 Secondary Control

In section 2.3 it is described, that the main advantage of secondary controlled system is that it is possible to recover some of the energy invested. Hence, disregarding energy losses from non-conservative forces such as friction, input energy are expected to be recoverable. As a consequence, the control design should not be penalising high amplitude input signals as long as it is within the physical limits of the system, i.e. maximum force output. However, as described in section 5.5, losses are associated with a change of input. This suggests a control objective of little or no overshoot or other control input oscillations.

7.1.2 Control Input Type

As described, the DDC can give 27 force steps distributed with some symmetry around zero. The steps are a result of the different combinations of pressure lines and cylinder chambers. Each chamber can be connected to one of three pressure lines, and as there are three chambers, this yields $3^3 = 27$ different combinations. The articles reviewed in the state of the art chapter on secondary control of multi-chamber cylinders all use some formulated logic in choosing the required valve combinations in order to achieve a desired force. The simplest logic is a minimum error, where the logic chooses which force output level closest to the force reference. This is also known as linear quantisation. More advanced logic penalises switching based on energy costs and impose minimum switching periods. Hence, all uses control logic to reformulate the multiple input-single output (MISO) control problem into a single input-single output (SISO) control system.

The MISO system can be seen as three chambers each generating an one of three force levels, whereas the SISO-approach generates a single force. While the SISO-approach enables generation of a single force reference by e.g. PI-control the logic implemented puts a certain bias on the output. The results of a combination of PI-control and advanced logic can thus be hard to interpret, and it makes the design of control parameters difficult. It is then up to the control engineer to balance the PI-control gains to support the purpose of the logic , e.g. minimising energy costs, while reducing tracking error.

7.1.3 Pressure Oscillations

The DDC suffers from large pressure oscillations when switching between pressure lines, but it is not possible to increase dampening by applying pressure feedback as for a regular proportionally controlled servo system. Instead, as discussed in [15], it is possible to design open loop opening area trajectories to reduce oscillations. However, this is done utilising the many DFCU valves during switching and thus not directly a part of the secondary control problem. Thus, the problems can be treated separately as long as the principles of secondary control are not affected (no continuous throttling). Hence, while the problem of large pressure oscillations can be addressed by avoiding frequent switching between pressure lines, increased system damping cannot be achieved with secondary control, but must be handled with other tools.

7.1.4 Effects of Quantization

For a regulation problem of stabilising a system at some equilibrium point, the quantization of the input can result in oscillations around the equilibrium point or limit cycles. Such behaviour can not be predicted by linear theory. Instead describing functions can be used to predict limit cycles. Describing functions are approximations of the frequency response function $N(A,\omega)$ of a non-linear term, that can be used to predict limit cycles of the form $x(t) = A_{lc}\sin(\omega_{lc}t)$, where A_{lc} and ω_{lc} is the amplitude and frequency of the limit cycle, respectively

To illustrate the problem, a unity feedback controller is implemented to drive the system to the origin. The system is chosen as a decoupled MISO system, where only the smallest cylinder chamber $V_{c3'}$ is considered. This is chosen in order to simplify the analysis. Thus, the system can be described with the following state equations, where $x_1 = x_s$, $x_2 = \dot{x}_s$ and $x_3 = p_{c3'}$.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{p}_{c3'} \end{bmatrix} = \begin{bmatrix} x_2 \\ (-B_v x_2 - \tanh\left(\frac{x_2}{\gamma}\right)\left(F_{C1} + F_{C2}e^{-|x_2|/k_{stri}}\right) + A_{c3'}p_{c3'})\frac{1}{m_s} \\ \frac{\beta_e(p_{c3'})}{V_{c3'} + A_{c3'}x_1}\left(Q_{c3'}(u) - A_{c3'}x_2\right) \end{bmatrix} \tag{7.1}$$

Here, $Q_{c3'} = n_3 A_0 c_d \sqrt{\frac{2}{\rho}\left(p_S - p_{c3'}\right)}\text{sgn}(p_S - p_{c3'})$ is the input flow, where p_S can be either of the three pressure line levels. A condition of describing function analysis is that there is only a single non-linear component [24]. Thus, the system is linearised in order to neglect other non-linear terms such as Coulomb friction and valve flow non-linearity. The state equations are linearised assuming the chamber volume is constant, i.e $V_{c3'} = V_{0c3'} = cst.$, thus yielding the state space model

$\dot{\mathbf{x}} = \mathbf{A}_{DDC}\mathbf{x} + \mathbf{B}_{DDC}Q_{c3}(u)$ and $y = \mathbf{C}_{DDC}\mathbf{x}$, where

$$\mathbf{A}_{DDC} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\dfrac{B_{vs}}{m_s} & \dfrac{A_{c3'}}{m_s} \\ 0 & -\Lambda A_{c3'} & 0 \end{bmatrix}, \quad \mathbf{B}_{DDC} = \begin{bmatrix} 0 \\ 0 \\ \Lambda \end{bmatrix} \tag{7.2}$$

$$\mathbf{C}_{DDC} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.3}$$

The eigenvalues of A_{DDC} are negative, and thus the system is stable. The open loop transfer function is then

$$G_{DDC}(s) = \frac{x_s}{Q_{c3'}} = (s\mathbf{I} - \mathbf{A}_{DDC})^{-1}\mathbf{B}_{DDC} \tag{7.4}$$

$$= \frac{1}{s}\frac{\Lambda A_{c3'}}{m_s s^2 + B_{vs}s + \Lambda A_{c3'}^2} \tag{7.5}$$

Here, $\Lambda = \dfrac{\beta_0}{V_{0c3'}}$, and the chosen linearisation point is at half stroke length. The non-linear effects of this quantizing of inputs can be compared to that of a relay of on-off non-linearity for the single valve manifold. Thus, near the origin, the closed loop control will only have two inputs to choose from, yielding positive and negative acceleration respectively. In [24], the describing function of a relay-type non-linearity is found to be $N(A_{lc}) = \frac{4M}{\pi A_{lc}}$, thus the describing function is only a function of the amplitude of the assumed limit cycle. M is the magnitude of the output, which in this case is $Q_{c3'}$. The input flow is modelled as the maximum flow when switching between medium and low pressure line and vice versa, where

$$Q_{c3'} = \begin{cases} n_3 k_v \sqrt{(p_m - p_l)}\,\mathrm{sgn}(p_m - p_l) \\ n_3 k_v \sqrt{(p_l - p_m)}\,\mathrm{sgn}(p_l - p_m) \end{cases} \implies Q_{c3'} = \pm Q_{max} \tag{7.6}$$

Thus, the input flow is constant positive or negative Q_{max} for a given state of the on-off relay. A block diagram of the system is shown in figure 7.1.

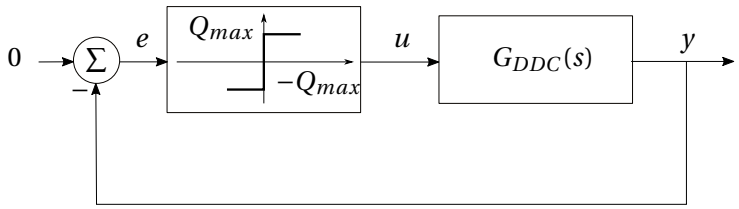


Figure 7.1. Block diagram of the describing function with $M = Q_{max}$ in series with the linear system.

If a limit cycle exists, it must satisfy the equation [24],

$$G_{DDS}(j\omega) = -\frac{1}{N(A_{lc})} \tag{7.7}$$

Since $N(A_{lc})$ is real for all A_{lc}, there will be a limit cycle if the Nyquist plot of $G_{DDC}(j\omega)$ crosses the negative real axis. Thus, we find the frequency of where $G_{DDC}(j\omega)$ is real.

$$\mathrm{Im}\left[G_{DDC}\left(j\omega_{lc}\right)\right] = 0 \tag{7.8}$$

$$\mathrm{Im}\left[\frac{1}{\left(j\omega_{lc}\right)}\frac{\Lambda A_{c3'}}{m_s\left(j\omega_{lc}\right)^2 + B_{vs}\left(j\omega_{lc}\right) + \Lambda A_{c3'}^2}\right] = 0 \tag{7.9}$$

$$\implies \omega_{lc} = A_{c3'}\sqrt{\frac{\Lambda}{m_s}} \tag{7.10}$$

Thus, indeed there is a limit cycle with approximate frequency $\omega_{lc} \approx 36$ rad/s. The amplitude can be found by evaluating

$$G_{DDS}\left(j\omega_{lc}\right) = -\frac{1}{N(A)} = -\frac{\pi A_{lc}}{4Q_{max}} \tag{7.11}$$

$$-G_{DDS}\left(j\omega_{lc}\right)\frac{4Q_{max}}{\pi} = A_{lc} \tag{7.12}$$

$$\implies A_{lc} = -\frac{4Q_{max}A_{c3'}B_{vs}\Lambda\omega_{lc}^2}{(m_s\omega_{lc}^3 - \Lambda A_{c3'}^2\omega_{lc})^2 + B_v^2\omega_{lc}^4} \tag{7.13}$$

The limit cycle then has an amplitude $A_{lc} \approx 0.15m$. The Nyquist plot of $G_{DDC}(s)$ and $N(A)$ can be seen in figure 7.2.

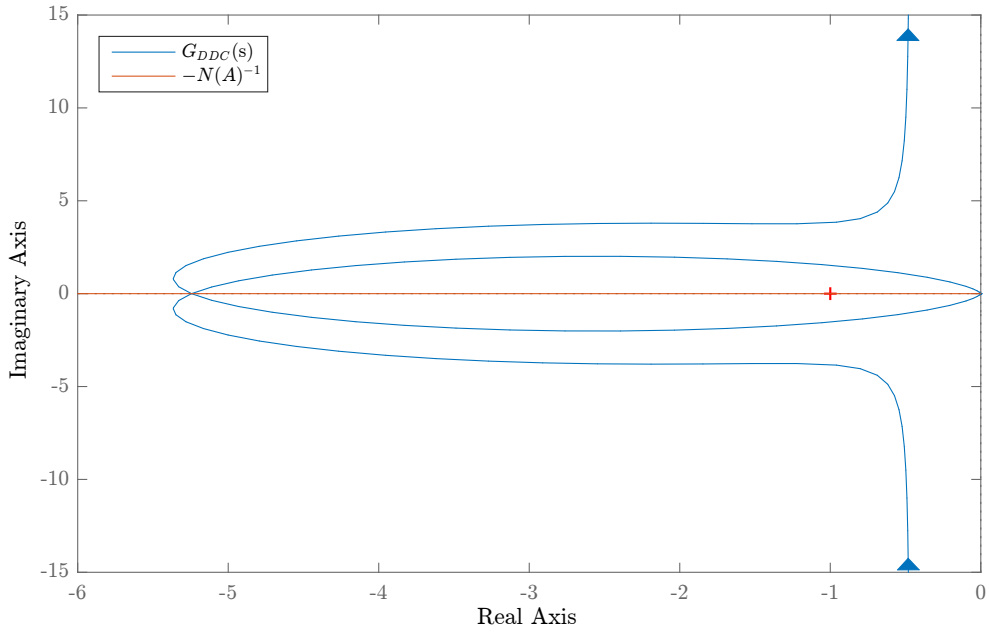


Figure 7.2. Nyquist plot of $G_{DDC}(s)$ and $-N(A)^{-1}$. $-N(A)^{-1} \to -\infty$ for $A \to \infty$ and $-N(A)^{-1} \to 0$ for $A \to 0$

According to the Limit Cycle Criterion in [24], the limit cycle is stable since the points of $-N(A)^{-1}$ trajectory for A increasing is not encircled by $G(j\omega)$ in the Nyquist plot. The values found for the limit cycle are highly approximate due to firstly the approximate nature of describing function and secondly the many approximations done, such as neglecting other non-linearities. However, insight is given into which parameters affect the amplitude and frequency. From the above analysis, it is

clear, that a smaller mass results in a larger limit cycle frequency and amplitude. Likewise, larger differences between line pressure increase limit cycle amplitude through Q_{max}. Conversely, larger friction terms result in smaller amplitudes. A similar analysis can be done for the SISO system with a linear quantizer. In figure 7.3 the gain of a linear quantisation for the 27 force levels of the DDC can be seen as a function of the input. As the force level steps does not include zero, the amplitude is negative infinite at zero. At higher inputs the relative quantization error decreases and the gain tends to unity constant gain, hence the name 'linear quantizer'.

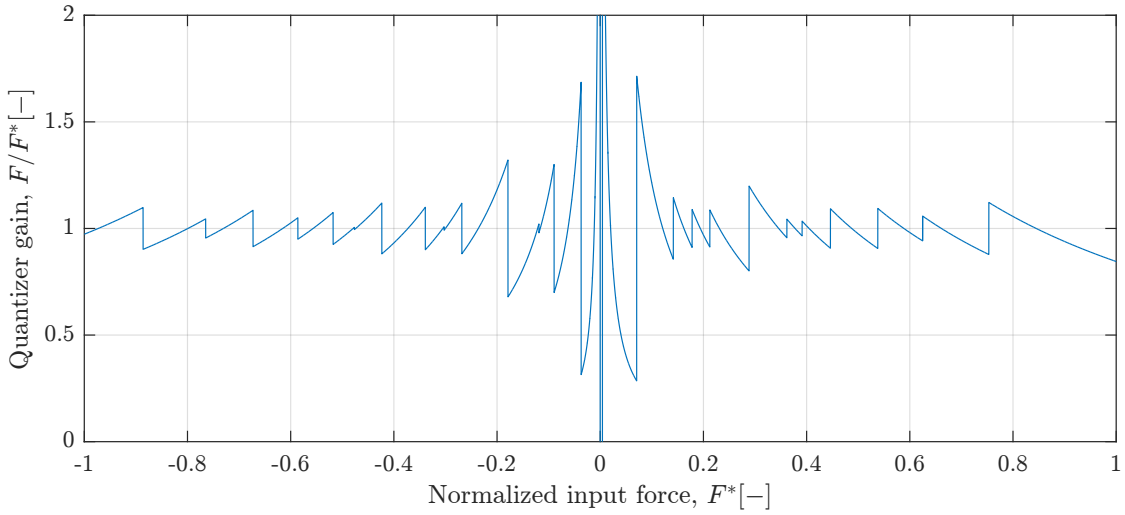


Figure 7.3. The input dependent gain of a linear quantizer on the DDC with 27 force steps.

Figure 7.4 shows the gain of linear quantisation on the DDC where the fourth chamber is utilised. This yields 81 force steps. It can be seen, that while the gain is still infinite at zero input, the gain tends to unity faster for larger inputs. Thus, a higher resolution of quantisation leads to a more smooth system input. If the force reference range is small, the relative quantization error is large. Thus, the system must be dimensioned such that the force required matches the force available. For a low mass system, the quantisation range is not properly utilised, leading to coarse force steps to the system. Figures 7.3 and 7.4 underline a fundamental difficulty in controlling quantised input systems and particularly a DDC in that, for a given operation point load holding is difficult due to the quantisation error yielding a non-zero acceleration.

To verify the existence of limit cycles in the system, the system is simulated with a P position controller. The P-gain are chosen such that it yields a stable system with approximately 5% overshoot. The results are shown in figures 7.5 and 7.6, for a linear quantization and an algorithm with a minimum switching period respectively. The minimum period is 0.3 seconds and a constant disturbance force of $130kN$ is added in both cases.

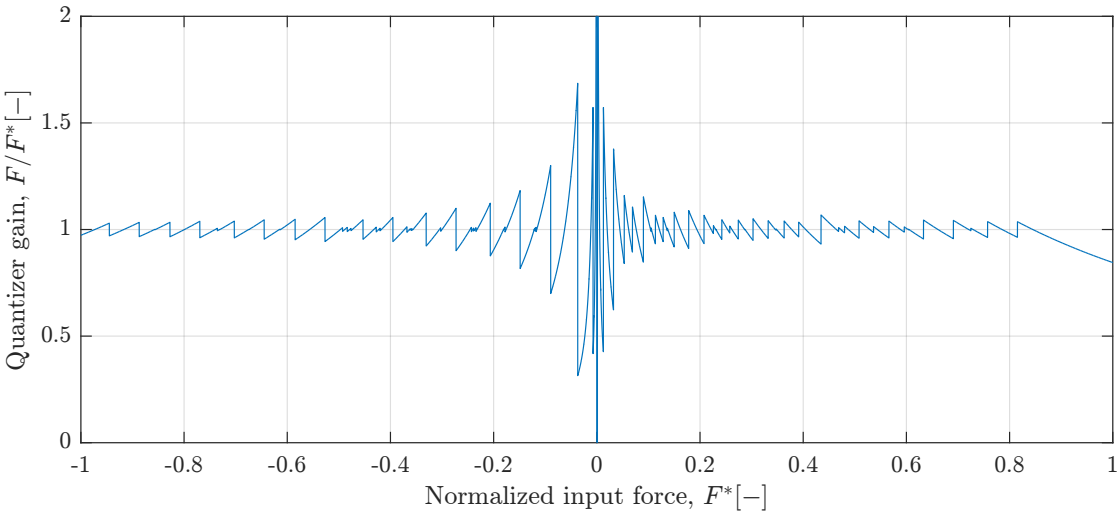


Figure 7.4. The input dependent gain of a linear quantizer on the DDC with 81 force steps.

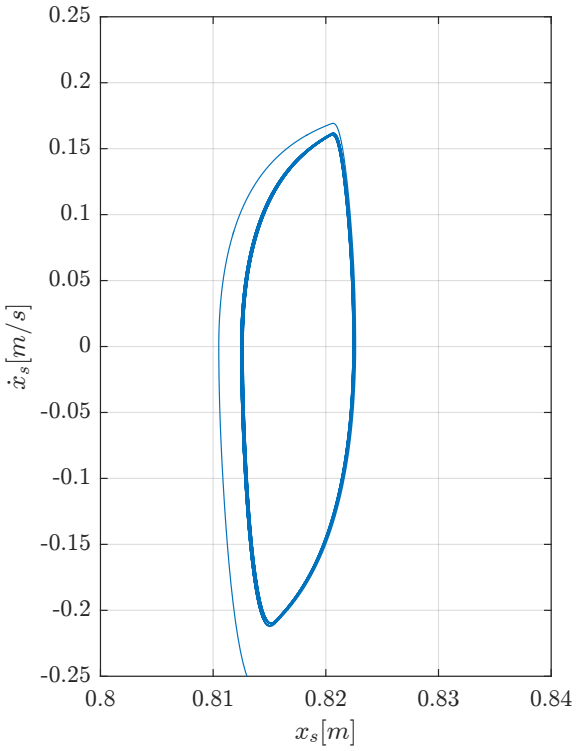


Figure 7.5. Simulated limit cycle with linear quantization.

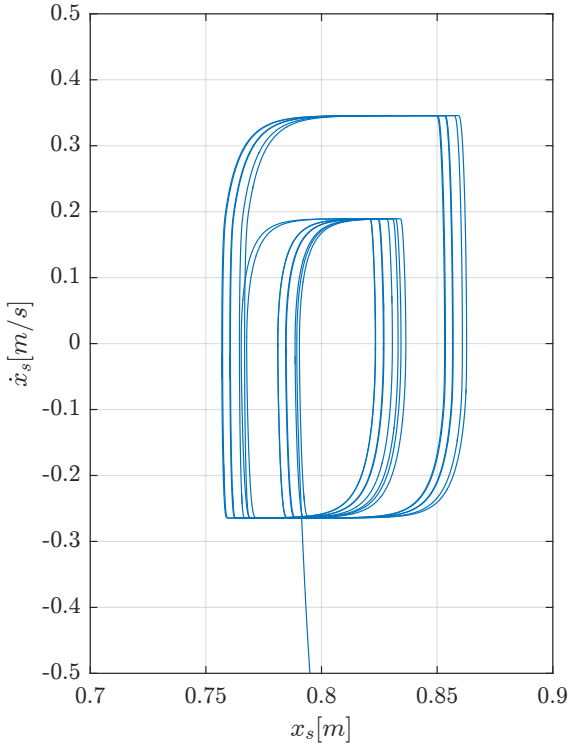


Figure 7.6. Simulated limit cycle with a minimum switching period quantizer.

The limit cycle in figure 7.5 has an amplitude of $0.005m$ and a frequency of approximately $10Hz$, while the frequency limited has an amplitude of approximately $0.05m$ and a frequency of $3.3Hz$. Thus, for quantized input there is a trade-off between the costs of switching and precision.

7.2 Application Case

To evaluate the control of the DDC it would be ideal to test it on data from a real life application in order to see the true potential of the controllers developed. Since it have not been possible to retrieve any real data for this thesis, a simple imaginary application case for the DDC is established to add extra complexity in the kinematics of the mechanics. The application is designed to match the sizes and dimensions of the DDC cylinder described in the modelling section 5.1.

The application established here is a reversed pendulum, where the DDC is mounted directly at the rod. A sketch is shown on figure 7.7. To simplify the system, the centre of mass is assumed to be in the middle of the mass at the end of the rod, thus the mass of the rod is neglected. The origin is defined as center of rotation which is located at the button of the rod connected to the ground.

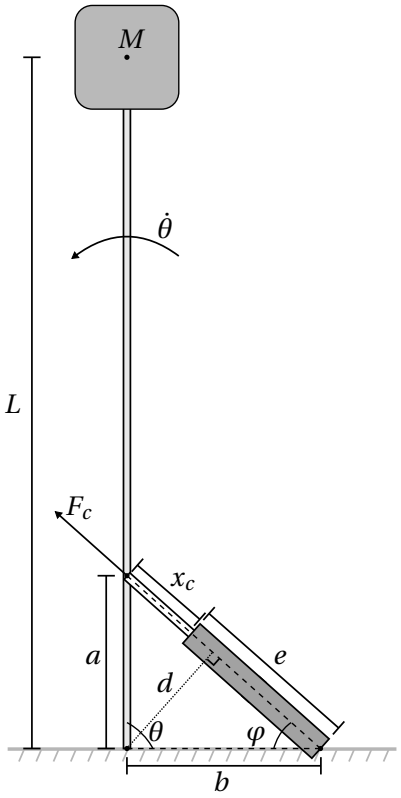


Figure 7.7. Sketch of application used to test controller performance.

M	$2 \cdot 10^3$ [kg]
L	8 [m]
a	2 [m]
b	2.236 [m]
e	2 [m]
x_c	{0 : 2} [m]
θ	{56 : 141.5} [degree]
d	{0.7 : 1.85} [m]

Table 7.1. Parameters used in application.

The kinematics of the application are here deducted. The lengths, a, b, e and L, and the mass M are all known constants. And x_c is a variable of the length that the cylinder is extracted. The angle, θ, is dependent on x_c and it is orthogonal when the cylinder is in centre position. The length, d, is the length from the origin to a point where d is orthogonal on the cylinder. The analytic expression for θ, φ and d is expressed in equation 7.16. Sizes and dimensions are listed in table 7.1.

$$\theta = \cos^{-1}\left(\frac{a^2 + b^2 - (x_c + e)^2}{2ab}\right) \tag{7.14}$$

$$\varphi = \cos^{-1}\left(\frac{(x_c + e)^2 + b^2 - a^2}{2(x_c + e)b}\right) \tag{7.15}$$

$$d = \sin(\varphi)b = b\sqrt{1 - \left(\frac{(x_c + e)^2 + b^2 - a^2}{2(x_c + e)b}\right)^2} \tag{7.16}$$

Newtons 2'nd low of rotation is used to describe the mechanics of the application, seen in equation

7.18. The friction used is based on the measured friction shown in appendix A.1 on page 125. The coulomb and stribeck values are the same as in table A.1, and the viscous damping is chosen as; $B = 9 \cdot 10^5 Nm/s$. Thus the friction formula is again expressed as

$$F_{fric} = \tanh\left(\frac{\dot{x}_s}{\gamma}\right)\left(F_{C1} + F_{C2}e^{-|\dot{x}_s|/k_{stri}}\right) + B\dot{x}_s \tag{7.17}$$

The effect of the gravitational force on the system is seen on figure 7.8, where the system extrema are illustrated. It is seen that gravitational torque changes direction around the centre position where there is no gravitational torque. For a pendulum the mass moment of inertia is defined as; $J = L^2 M$.

$$J\ddot{\theta} = \tau_c - \tau_{fric} - \tau_g \tag{7.18}$$

$$\begin{aligned} \tau_c &= F_c d \\ \tau_{fric} &= F_{fric} d \\ \tau_g &= F_g L \cos(\theta) \\ F_g &= Mg \end{aligned} \tag{7.19}$$

Where g is gravitational acceleration.

Since the objective is control of the cylinder, the rotational expression above is transformed into a mass equivalent force expression with respect to the cylinder.

$$M_{eq}(x_c)\ddot{x}_c = F_c - F_{fric} - F_{g,eq}(x_c) \tag{7.20}$$

Where $M_{eq}(x_c)$ and $F_{g,eq}(x_c)$ are the equivalent mass and equivalent gravitation respectively. They are both a function of cylinder position and are expressed in equation 7.22.

$$M_{eq}(x_c) = \frac{J}{d(x_c)^2} = \frac{L^2 M}{d(x_c)^2} \tag{7.21}$$

$$F_{g,eq}(x_c) = \frac{F_g L \cos(\theta(x_c))}{d} \tag{7.22}$$

Furthermore, the equivalent loads in the cylinder are shown in figure 7.9 and 7.10 as a function of cylinder position.

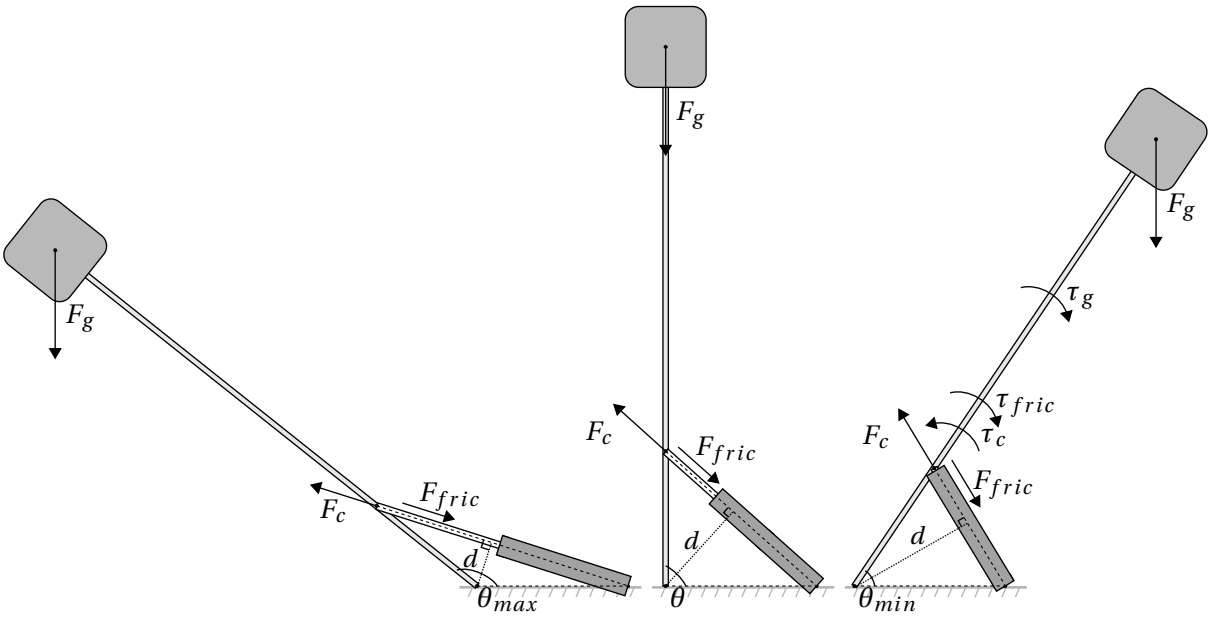


Figure 7.8. Maximum, central and minimum extraction of cylinder.

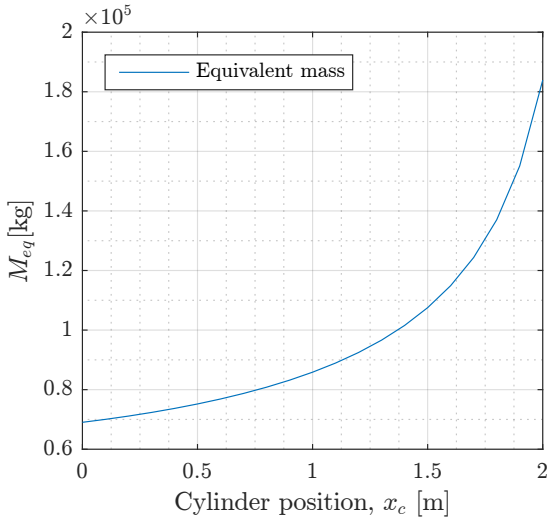


Figure 7.9. Equivalent mass on cylinder as function of cylinder position.

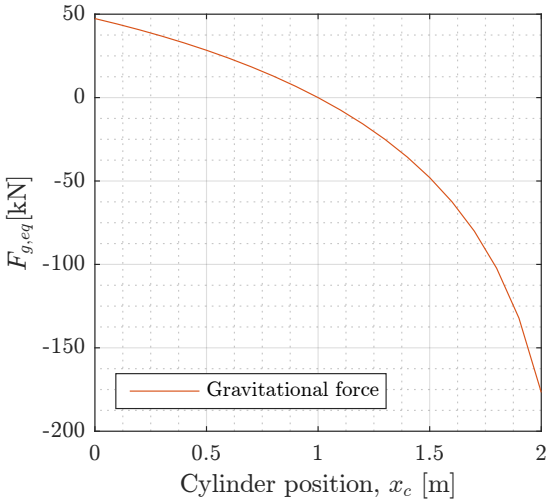


Figure 7.10. Equivalent gravitational force acting on cylinder as functon of cylinder position.

This application case is chosen such to test control in several conditions. While not itself necessarily a realistic set up, it gives the opportunity to test in conditions that are challenging control problems such as overrunning loads, low and high mass systems as well as load holding for different static loads.

7.3 Chapter Summary

This chapter described some fundamental characteristics of secondary control of a discrete displacement cylinder. It was concluded, that it was not feasible to attempt to dampen the pressure oscillations inherent to a DDC system with a secondary control strategy. Furthermore, it was shown how the input quantisation associated with the DDC can induce limit cycles, and a relation between physical parameters and the limit cycle frequency and amplitude was found.

Finally, an application case suitable for position and velocity control was introduced.

The next chapter describes the general ideas behind Model Predictive Control and some of the main strengths and weaknesses of this control type.

CHAPTER 8

MODEL PREDICTIVE CONTROL

This chapter outlines the principles of Model Predictive Control and some considerations on the tuning variables and their effect on the control problem. For more information on Model Predictive Control, see [25] and for explicit Non-linear Model Predictive Control, see [19].

As mentioned, the scope of this thesis is to design a control topology suited to compromise the relation between the system input energy and a tracking performance. The control problem in this thesis is designed to regulate a constant or time varying reference which is only know at the exact time instance and thus unknown for future time steps. The focus of the controllers developed in this thesis are based on regulating velocity and position.

The basic idea of Model Predictive Control (MPC) is to use a model of the controlled system to predict future outputs on a determined time horizon as a function of the current measured state x and future inputs u. The future inputs are then optimised with respect to some chosen criteria, e.g. tracking error. The output is then applied until new state information is available, and optimal future inputs are calculated anew. An advantage of MPC is the possibility to include physically and operationally motivated constraints on state variables and inputs, both on magnitudes and rates of change, in the optimisation problem. For this thesis the focus is on a Model Predictive Control problem for a constrained non-linear system, referred to as Non-linear Model Predictive Control (NMPC). Another advantage lies in the predictive power of the model as e.g. system delays and other non-linearities can be compensated for as long as the model is sufficiently accurate. Often, discrete time models are used in MPC. Thus, in general a non-linear system can be described as

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{8.1}$$

$$y(t) = \mathbf{C}\mathbf{x}(t) \tag{8.2}$$

Estimating the future states using the model, the estimate is denoted $\mathbf{x}(t+k+1|t)$ to reflect that only state knowledge from time t is used in the estimation and $k = 0, 1, 2..., N$, where N is the time horizon. The future inputs are then $\mathbf{u}(t+k+1)$. A general optimisation problem can then be described as

$$V^*(\mathbf{x}(t)) = \min_{\mathbf{U}} J(\mathbf{U}, \mathbf{x}(t)) \tag{8.3}$$

subject to $\mathbf{x}(t|t) = \mathbf{x}(t)$ and

$$\mathbf{u}_{min} \leq \mathbf{u}(t+k) \leq \mathbf{u}_{max} \tag{8.4}$$

$$g(\mathbf{x}(t+k|t), \mathbf{u}(t+k)) \leq 0 \tag{8.5}$$

$$\mathbf{x}(t+k+1|t) = f(\mathbf{x}(t+k|t), \mathbf{u}(t+k)), \quad k \geq 0 \tag{8.6}$$

Here, $\mathbf{U} = [\mathbf{u}(t), \mathbf{u}(t+1), \mathbf{u}(t+2), ..., \mathbf{u}(t+N-1)]$ is the set of control moves. The problem solution is denoted $\mathbf{U}^* = [\mathbf{u}^*(t), \mathbf{u}^*(t+1), \mathbf{u}^*(t+2), ..., \mathbf{u}^*(t+N-1)]$ with corresponding cost function value V^*. g is a set of general inequality constraints, which can be physically motivated or stability-preserving from a solution point of view. The cost function can be described as

$$J(\mathbf{U}, \mathbf{x}(t)) = \sum_{k=0}^{N-1} [\ell(\mathbf{x}(t+k|t), \mathbf{u}(t+k))] + S(\mathbf{x}(t+N|t)) \tag{8.7}$$

ℓ is known as the stage cost, and S is known as the terminal cost. In general, they both affect control performance and stability, and are central to achieving good results. Once a solution is found, $\mathbf{u}(t)$ is applied to the system according to the receding horizon principle, until the next sampling instant $(t+k)$, where $\mathbf{U}$ is re-optimised using new state information $\mathbf{x}(t+k|t+k) = \mathbf{x}(t+k)$. Thus, MPC is in general a form of closed loop control, even though the solution to optimisation problem is open loop. The horizon is normally chosen such that the system settling time is within the time horizon [26]. However, increasing N also increases the size of the optimisation problem.

The above problem could describe a control objective to drive the system to the origin, but could also easily be adapted to a reference tracking problem $y(t) - r(t)$. Figure 8.1 show the structure of the MPC optimization for one iteration for the whole horizon N.

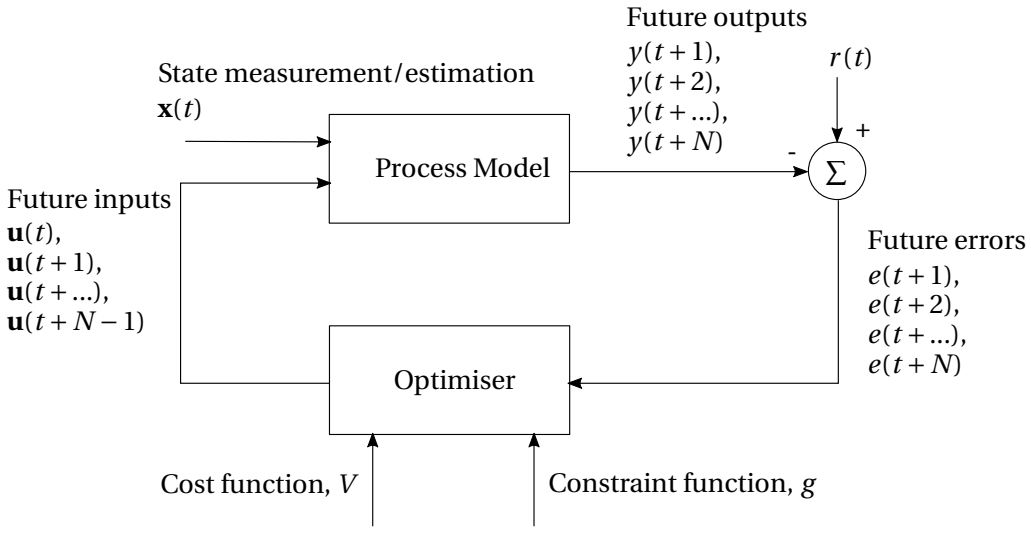


Figure 8.1. Structure of MPC for one iteration on horizon N.

The optimisation problem is a multi-parametric Non-Linear Programming problem (mp-NLP problem) parameterised in $\mathbf{x}$. Hence, it belongs to non-linear branch of MPC (NMCP), which is the focus in this thesis. Similar problem formulations exists in linear MPC, described as convex quadratic programming (QP) problems.
When using a nominal model, it is also possible to account for model uncertainties and measurement errors in the problem formulation. This is not considered further in this report.

8.1 Cost Function

Solutions to an optimal control problem are based on minimising an appropriate cost function for the system. Thus, optimal control is not necessarily perfect, but it is optimal in the sense of finding the best solution to a given cost function with some constraints. Therefore, a well chosen cost function is a critical part of ensuring good control performance.

In MPC the form of the cost function to be minimised can be seen in the form of equation 8.8 [19].

$$J(\mathbf{x}(t),\mathbf{u}(t)) = \sum_{k=0}^{N-1} \ell\big(\mathbf{x}(t+k|t),\mathbf{u}(t+k)\big) + S\big(\mathbf{x}(t+N|t)\big) \tag{8.8}$$

where ℓ is stage cost, S is the terminal cost, N is the horizon and $\mathbf{x}(t+k|t)$ is the system states at time $t+k$ given the knowledge of system at time t. Included in the stage cost is an expression for penalising system states and input. The terminal cost is then used to include a penalty on the last point in the horizon.

For a reference tracking problem, where $y(t)$ needs to follow a reference $r(t)$ and all states, $\mathbf{x}(t)$ are obtainable by measurement or estimation, the cost function can be expressed as in equation 8.9. The terms of the stage and terminal cost are expressed as quadratic norms with the constant weighting matrices, Q, R and P.

$$J\Big(\mathbf{x}(t),r(t),\mathbf{u}(t)\Big) = \sum_{k=0}^{N-1} \Big(\big\| \ell_x\big(x(t+k|t),r(t|t)\big)\big\|_Q^2 + \big\| h\big(\mathbf{x}(t+k|t),\mathbf{u}(t+k)\big)\big\|_R^2 \Big) + \big\| S\big(x(t+N|t),r(t|t)\big)\big\|_P^2 \tag{8.9}$$

where ℓ_x is a state cost and h is input cost as function of system states and input. Q refers to the penalty on system states. R is the input penalty. P penalises the terminal cost. Furthermore Q,R,P are positive semi-definite. $\|\bullet\|_A$ is the A-weighted 2-norm.

The trade-off between the weighting variables, Q and R, is based on how much the system input energy is penalised compared to the tracking error. Some considerations when choosing the Q and R weights have to be made in order to ensure stability. Conventional LQR tuning can be a helpful guideline to find an appropriate starting point [19]. In [19] MPC is designed for two different systems with quantized input, the terminal cost is in both cases a weighted quadratic norm, where the weighting coefficient, P, is equal to the coefficient Q. Thus, this could be a starting point when designing the cost function.

In order to have an intuitive relation between the weighing variables it can be beneficial to normalise the terms of the cost function. Generally a normalisation of the gradient can ease the convergence of the equation, because the values of all terms will be in the same range.

8.2 Tuning and Stability

In [27] one of the objectives is to determine some essential parameters that are used to obtain stability. The authors base their research on a large number of article regarding stability conditions for MPC. The parameters that are found useful in developing stable MPC with finite horizon are the terminal cost S and the terminal constraint set $\mathbb{X}_T$. The terminal constraint set determines the set in which the output state has to be within in the last horizon step, in order to obtain a feasible solution.
The terminal cost is an expression for the penalty of the last state in the horizon. Ideally

the terminal cost should be equal to the infinite horizon cost function, i.e $S(\mathbf{x}(t+N|t)) = \sum_{k=N}^{\infty}[\ell(\mathbf{x}(t+k|t),\mathbf{u}(t+k))]$.The cost function value corresponding to an infinite horizon would then be obtained and stability would automatically follow [27]. However, it is generally hard to compute and simple quadratic approximations are usually chosen [19].

8.3 Explicit Model Predictive Control

A disadvantage of MPC, especially NMCP, is that large computational power is required to solve the above problem in real time at each sampling interval. For linear convex QP problems very efficient numerical methods exist, and linear MPC are well suited for processes with slow dynamics, such as chemical process control, with tens or hundreds of state variables, but with sampling times in order of minutes.[28] However this is not the case for NMPC, where challenges lies in achieving sufficiently good results to the mp-NLP problem in real time. One way to address this problem is found in explicit MPC. Explicit MPC (eMPC) builds on the idea that it is possible to construct piecewise linear (affine) state feedback functions to approximate the a solution to a mp-NLP. The solution is then found offline and as an explicit function of the state vector. Thus online, on the real-time computer, the implemented eMPC maps into a lookup table, and then requires only few simple evaluations or comparisons to determine the approximate optimal input.

8.4 Chapter Summary

This chapter introduced the general idea behind Model Predictive Control. The idea is to utilise the predictive power of a model to determine an optimal series of control actions, based on desired system behaviour. The desired behaviour is then described by the cost function, which is a function of system input and output. Some general guide lines for tuning and stability was presented. Lastly, the idea of explicit Model Predictive Control was introduced.

The next chapter concerns the development of a Non-linear Model Predictive Control Controller for the application case described in section 7.2.

CHAPTER 9

DESIGN OF NON-LINEAR MODEL PREDICTIVE CONTROL

This chapter describes the application of NMPC on the application case using a DDC as an actuator. In the first part, the optimisation problem is formulated. This includes describing a reduced order model suitable for NMPC. Furthermore, the cost and constraint functions are discussed, including criteria on energy efficiency and control performance, and the use of terminal constraints and terminal costs. Finally a summary of the NMPC problem is made together with a discussion of control parameters, followed by results showing the control performance.

9.1 Reduced Order Discrete Time Model

In order to have a tractable optimisation problem, a reduced order model is used for describing the system dynamics in the optimisation problem. It is important to consider the dynamics that has to be included in the model. Two reduced order models are considered: a second order model, including only the mechanical system described by Newton's laws of motion and fifth order model including the pressure build up in each cylinder chamber. From the discussion in section 7.1.3, the purpose of the control is mainly position or velocity control and it cannot be expected to dampen pressure oscillations by control. Furthermore, the pressure dynamics where seen to be in the order of milliseconds and much faster than the mechanical system to be controlled.

Resultantly, the choice has fallen on a second order mechanical model, where pressure build up have been neglected. However, pressure level switching may still be penalised based on pressure levels as described later in this chapter. The mechanical model is described as in section 5.3

$$M_{eq}(x_c)\ddot{x}_c = F_c - F_{fric} - \frac{MgL\cos(\theta(x_c))}{d} \tag{9.1}$$

In order to use the model for NMPC, it is converted into a discrete time state model. Given that pressure build up are neglected the model is fairly well conditioned. Thus, it is considered reasonable to perform the numerical integration for the explicit forward Euler method with time step T_s. This

yields a discrete time model given as

$$x_1(t+T_s) = x_1(t) + T_s\,[x_2(t)] \tag{9.2}$$

$$x_2(t+T_s) = x_2(t) + T_s\left[\left(-F_{fric} - \frac{MgL\cos(\theta(x_1(t)))}{d}\right)M_{eq}(x_1(t))^{-1} + F_c(\mathbf{u}(t))M_{eq}(x_1(t))^{-1}\right] \tag{9.3}$$

$$y(t) = \mathbf{C}_j\mathbf{x}(t) \tag{9.4}$$

where $C_1 = \text{diag}[1,0]$ for position control and $C_2 = \text{diag}[0,1]$ for velocity control. In general for this problem subscript j is used to distinguish between position and velocity control for $j = 1$ and $j = 2$ respectively. The input force is described as $F_c(\mathbf{u}(t))$, where input vector is $\mathbf{u} = [u_1, u_2, u_3]$, i.e. one input for each cylinder chamber, numbered appropriately. Thus, the control approach chosen is the MISO approach. This is done in order to avoid any bias imposed by introducing switching logic and instead use penalty and constraints to define the control objective.

$$F_c(\mathbf{u}) = -F_{c1}(u_1) + F_{c2}(u_2) + F_{c3}(u_3) \tag{9.5}$$

Each input can take one of three values, from 1 to 3. Each value correspond to a pressure level command in the corresponding chamber.

$$\left.\begin{array}{l} u_i = 1 \implies F_{ci}(u_i) = p_h A_{ci} \\ \quad = 2 \implies F_{ci}(u_i) = p_m A_{ci} \\ \quad = 3 \implies F_{ci}(u_i) = p_l A_{ci} \end{array}\right\} \quad \text{for} \quad i = 1,2,3 \tag{9.6}$$

Thus, each u_i results in a force acting on the mechanical load and F_c is the sum of forces from each cylinder chamber acting on the mechanical load. The reason for this indexing of inputs is due to the choice of numerical optimiser, see appendix B.

9.2 Cost Function

The cost function seen in equation 8.9 consist of input cost, state cost and terminal cost. For the DDC the input cost can effectively be seen as energy loss for switching from a previous pressure state in a chamber to a new pressure state given by the input. In section 5.5 a detailed description of the system losses is presented.

The only loss directly dependent on input is the switching loss, which is described as the difference between supplied energy to chamber and the potential pressure energy saved in the compression. The throttling losses are only a function of x_2 as was seen in section 5.5. Thus, in order to simplify the expression for the cost function the throttling losses is not penalised directly, but will instead be penalised through the penalty of the velocity state x_2. The energy losses for the secondary controlled system described in section 5.5, does not include energy loss associated with input magnitude, because input energy is expected to be recoverable for a secondary controlled system. However, in order to increase the controllability and tuning abilities for the NMPC a cost on input magnitude is introduced. The input magnitude is expressed as the expected force for the given input. Subsequently, a penalty on force results in a penalty on acceleration.

The function for input cost is the same for position and velocity control, hence the subscript j is omitted in the expression for the input cost, which is expressed as

$$\mathbf{h}\Big(\mathbf{x}(t),\mathbf{u}(t),\mathbf{u}(t-1)\Big) = \begin{bmatrix} \left(\sum_{i=1}^{3}\frac{1}{2}\Big(p_i\big(u_i(t)\big) - p_i\big(u_i(t-1)\big)\Big)^2\frac{V_i(x(t))}{\beta}\right)E_{\beta,max}^{-1} \\ \Big(-A_{c1}u_1(t) + A_{c2}u_2(t) + A_{c3'}u_3(t)\Big)F_{max}^{-1} \end{bmatrix} \tag{9.7}$$

Where V_i and p_i are the volume and pressure of the i'th chamber respectively. The pressure, p_i, is determined as a function of the input at the given time as shown in equation 9.8. It is worth noticing, that this implementation results in an "open loop" input cost, as the pressures that are penalised are not measured, but estimated steady state values based on $\mathbf{u}$. The constant $E_{\beta,max}$ is the maximum switching loss used to normalise the cost function and is derived in equation 9.9. F_{max} is the maximum available force.

$$\left.\begin{array}{ll} u_i & = 1 \Rightarrow p_i = p_l \\ & = 2 \Rightarrow p_i = p_m \\ & = 3 \Rightarrow p_i = p_h \end{array}\right\} \quad \text{for} \quad i = 1,2,3 \tag{9.8}$$

$$E_{\beta,max} = \frac{1}{2}(p_h - p_l)^2 \frac{L_s A_{c1} + V_0}{\beta} \tag{9.9}$$

Where L_s is the cylinder stroke length, A_{c1} is cross-sectional area of chamber 1 and V_0 is the initial volume for all chambers. The state cost is defined similar to equation 8.9 in section 8.1. In order to penalise the state cost individually for position and velocity control, it is denoted as a vector. For position control the velocity state is included in the state cost, in order to be able to penalise the velocity state and thereby the throttling losses. When velocity control is performed the position state is omitted in the state cost function. The function for state cost for position and velocity control are expressed as

$$\boldsymbol{\ell}_{x1}\big(\mathbf{x}(t+k|t), r(t|t)\big) = \begin{bmatrix} \dfrac{x_1(t+k|t) - r(t|t)}{e_{1,max}} \\ \dfrac{x_2(t+k|t)}{x_{2,max}} \end{bmatrix} \quad , \quad \boldsymbol{\ell}_{x2}\big(\mathbf{x}(t+k|t), r(t|t)\big) = \begin{bmatrix} 0 \\ \dfrac{x_2(t+N|t) - r(t|t)}{e_{2,max}} \end{bmatrix} \tag{9.10}$$

where $r(t|t)$ indicates a constant reference for the optimisation problem input at time t, i.e. a trajectory is assumed not available for this problem. The maximum state error $e_{j,max}$ used to normalise the state cost is defined as

$$e_{1,max} = L_s \tag{9.11}$$

$$e_{2,max} = 2x_{2,max} \tag{9.12}$$

The terminal cost, S_j, is similar to the state cost. In order to penalise the terminal cost individually for position and velocity, it is denoted as a vector. The expression for the terminal cost is seen in equation 9.13.

$$\mathbf{S}_1\big(\mathbf{x}(t+N|t), r(t|t)\big) = \begin{bmatrix} \dfrac{x_1(t+N|t) - r(t|t)}{e_{1,max}} \\ \dfrac{x_2(t+N|t)}{x_{2,max}} \end{bmatrix} \quad , \quad \mathbf{S}_2\big(\mathbf{x}(t+N|t), r(t|t)\big) = \begin{bmatrix} 0 \\ \dfrac{x_2(t+N|t) - r(t|t)}{e_{2,max}} \end{bmatrix} \tag{9.13}$$

The final cost function is expressed by rewriting equation 8.9 as

$$\begin{aligned} J_j\Big(\mathbf{x}(t), r(t), \mathbf{u}(t), \mathbf{u}(t-1)\Big) = \sum_{k=0}^{N-1} \Bigg(& \Big\| \boldsymbol{\ell}_{xj}\big(\mathbf{x}(t+k|t), r(t|t)\big) \Big\|_{Q_j}^2 + \Big\| \mathbf{h}_j\big(\mathbf{x}(t+k|t), \mathbf{u}(t+k), \mathbf{u}(t+k-1)\big) \Big\|_{R_j}^2 \Bigg) \\ & + \Big\| S_j\big(\mathbf{x}(t+N|t), r(t|t)\big) \Big\|_{P_j}^2 \end{aligned} \tag{9.14}$$

The subscript j on the weighting matrices Q, R and P denotes the difference in weightings for position and velocity control.

9.3 Constraints

The constraints on the system are mainly physically motivated. Thus, a hard physical constraint is the stroke length of the cylinder. As the reduced order model does not include the continuity and orifice equations, the pressure drop across the manifold is not included. In a sense, the actuator then has unlimited power available as input pressure is constant regardless of the flow. Throttling losses are penalised in the cost function, but a constraint on velocity can be desirable. Ultimately, the constraints are chosen as

$$\begin{matrix} 0 \leq x_1(t+k|t) \leq L_s \\ -0.5 \leq x_2(t+k|t) \leq 0.5 \end{matrix} \quad \text{for} \quad k = 1,..,N-1 \tag{9.15}$$

From a stability point of view the terminal constraint set should be chosen as small as possible [19]. However, due to the quantized input no equilibrium point at the reference may exist, but instead there might be an offset from the reference or a limit cycle about the reference. Thus, when determining the terminal constraint set the magnitude of the possible limit cycle should be taken into account, in order to obtain a feasible solution. An initial value for the terminal constraint set is then the magnitudes of the limit cycle found in section 7.1.4.
When position control is performed the terminal constraint on both position and velocity is taken into account. The terminal constraint on the velocity included to secure that the system is steady in the last horizon step. However, when velocity control is performed the terminal constraint on the position is omitted, because only the velocity is of interest. The terminal constraint set for position and velocity control is expressed in equation 9.16 and 9.17.

$$\left.\begin{matrix} -0.1 & \leq x_1(t+N|t) - r(t|t) & \leq 0.1 \\ -0.3 & \leq \quad x_2(t+N|t) & \leq 0.3 \end{matrix}\right\} \Rightarrow \mathbb{X}_{T1} = \left[-0.1, 0.1\right] \times \left[-0.3, 0.3\right] \tag{9.16}$$

$$\left.\begin{matrix} -0.3 & \leq x_2(t+N|t) - r(t|t) & \leq 0.3 \end{matrix}\right\} \Rightarrow \mathbb{X}_{T2} = \left[-0.3, 0.3\right] \tag{9.17}$$

9.4 Problem Summary and Tuning

Based on the reduced order model described in equations 9.2 and 9.3, the cost function in 9.14 and the constraints of 9.15, 9.16 and 9.17, the optimisation problem is

$$\begin{aligned} V^*(\mathbf{x}(t)) = \min_{\mathbf{U}} \sum_{k=0}^{N-1} \Bigg(& \left\| \frac{y(t+k|t) - r(t|t)}{e_{j,max}} \right\|_Q^2 + \left\| \mathbf{h}_j\big(\mathbf{x}(t+k|t), \mathbf{u}(t+k), \mathbf{u}(t+k-1)\big) \right\|_R^2 \Bigg) \\ & + \left\| S_j\big(\mathbf{x}(t+N|t)\big) \right\|_P^2 \end{aligned}$$

subject to state equations 9.2, 9.3, $\mathbf{x}(t|t) = \mathbf{x}(t)$ and

$$\begin{aligned} & u_i(t+k|t) \in \{1,2,3\} \quad , \quad k = 0,...,N-1 \quad , i = 1,2,3 \\ & \left.\begin{matrix} 0 & \leq x_1(t+k|t) & \leq L_s \\ -0.5 & \leq x_2(t+k|t) & \leq 0.5 \end{matrix}\right\} \quad k = 1,..,N-1 \\ & \mathbf{x}(t+N|t) \in \mathbb{X}_{Tj} \end{aligned}$$

9.4.1 Time Step and Horizon

The choice of forward Euler time step T_s and horizon N are chosen based on a balance between control performance and computation time. A minimum $T_{s,min} = 0.05s$ is imposed to ensure the validity of disregarding the pressure build i the cylinder chambers. This is based on the pressure build up in figure 5.3 and simulation data. From simulation of the full order model, the mechanical settling time for the velocity response varies with cylinder position between 0.45 and 1.5 seconds in the operating range of 0.5 to 1.5 m. Different combinations of horizon and time step was tested. It was found that a too large T_s resulted in a too large acceleration over the time period, thus limiting the feasible amount of force steps available. Likewise, a too large horizon was found to be increasing the dimension of the optimisation problem and hence increasing the computational time required to achieve a feasible result. Moreover, a small horizon with tight terminal constraints forces the control optimiser to apply a larger input force in order to obtain a feasible solution. Resultantly, a small T_s and N was found preferable, and testing yielded good results with $T_s = 0.1$, $N = 15$ for position control. For velocity control, the minimum step time, $T_s = T_{s,min}$ is chosen with a horizon $N = 15$.

9.4.2 Evaluation of Computational Parameters

The problem is solved using *glcSolve* from the TOMLAB optimization package for MATLAB. As is described in appendix B, the numerical global optimisation routine has no convergence parameters, as it is not gradient based. Instead, the amount of function evaluations are used to achieve a desired level of accuracy. This is a tuning variable as there is no explicit relation between number of function evaluations and accuracy. To determine the number of function evaluations that should be used, the solution to the cost function is evaluated as function of the number of function evaluations, which can be seen in appendix C. Based on the obtained results the number of function evaluations is chosen to be 4000 evaluations. Since the accuracy of the solution is based on the number of function evaluations the result can not be said to be a global optimum, but instead a 'close-to-global' optimum.

In appendix C the computational time of the developed NMPC position controller is evaluated. An average computational time for each time step for the NMPC position controller is calculated to be 0.23 seconds, 1.3 seconds and 9 seconds for 500, 3000 and 15000 function evaluations respectively. Thus, the developed NMPC controller cannot achieve desirable results in real time for this system. There is then a need for an approximate explicit optimal control law in order to reduce the required computational time.
Even though the NMPC controller cannot achieve desirable results in real time, it will be used to tune the control parameters, in order to use the control parameters for the eMPC controller that will be developed later. Furthermore, results from the NMPC position and velocity controller implemented on the full order model, will be used as comparison for the eMPC controller and the linear controller, that as well will be developed later.

9.4.3 Parameter Tuning for Position Control

The NMPC position controller is implemented in MATLAB on a simplified model of the system, without transmission line dynamics, in order to test and adjust the optimisation parameters, such as the weighting metrices for state, input and terminal cost. To evaluate how the controller is able to balance between energy losses and tracking error, the controller is first evaluated with a very limited input penalty. Figures 9.1 and 9.2 show the simulated results for position and force reference for position control with $R_1 = 10^{-6}\text{diag}[1, 0.0005]$ and $Q_1 = P_1 = \text{diag}[15, 0.1]$, initial conditions $(0.8, 0)$

and constant reference r = 1.1m. The right label of figure 9.2, shows the reference pressures in chamber c_1, c_2 and c_3 respectively. It can be seen that many small force shifts are made, instead of keeping a constant force level for longer time and thereby decrease the number of shifts. The many small force shifts result in simultaneous change in pressures in the chambers. Thus, the force shifts are not made energy efficient. Assuming the pressure in the chambers are the reference pressure, the energy losses for the step corresponds to approximately 27.4 kJ.

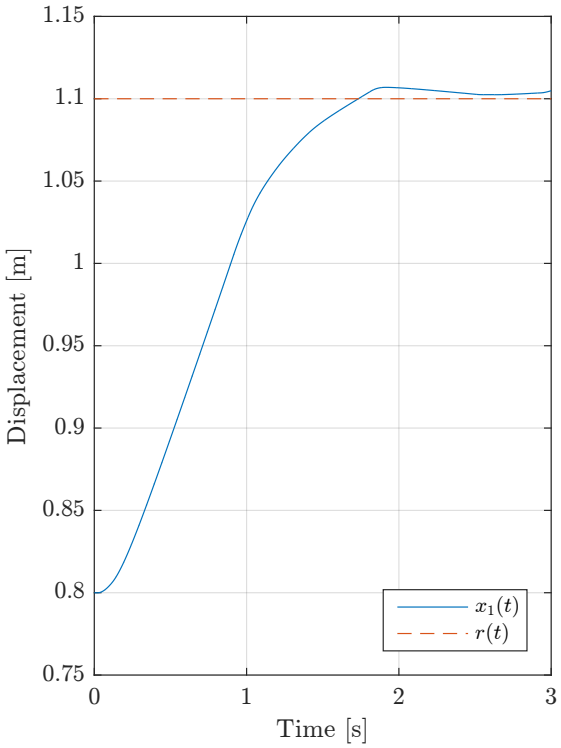


Figure 9.1. NMPC position controller performance with small input penalty.

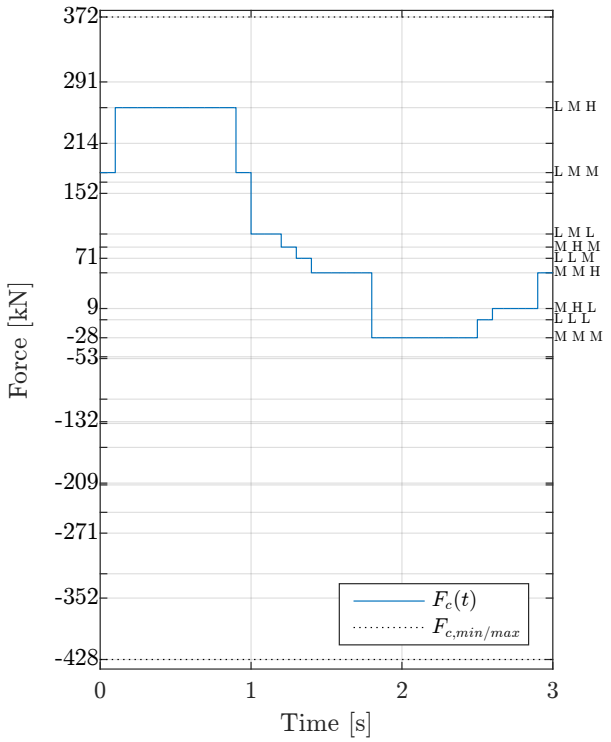


Figure 9.2. Force reference for NMPC position control with small input penalty.

In figure 9.3 and 9.4 the force shifts are seen to be reduced by increasing the input penalty to R_1 = diag[1, 0.0005], while keeping the other weightings at $Q_1 = P_1$ = diag[15, 0.1]. The position response does not change notably, but the controller generally chooses a force level for longer time and does not change all chambers simultaneously. Assuming the pressure in the chambers are the reference pressure, the energy losses for the step corresponds to approximately 7.4 KJ. Thus, the energy efficiency is increased significantly by increasing the input penalty. It should be noted that it is not the number of force shifts that are penalised, but the energy loss associated with the shifts. When the controller is close to the reference it generally chooses the force level corresponding to MMM, as it is "cheap" to switch from medium pressure to either high or low pressure. In equation 5.34 in section 5.5 it is illustrated that the energy loss associated with a shift $HMM \rightarrow LMM$ is larger than the energy loss from the sum of shifts $HMM \rightarrow MMM \rightarrow LMM$, and vice versa from low to high. Thus, with the predictive power of the controller the force level corresponding to MMM is chosen when a small or no force is needed.

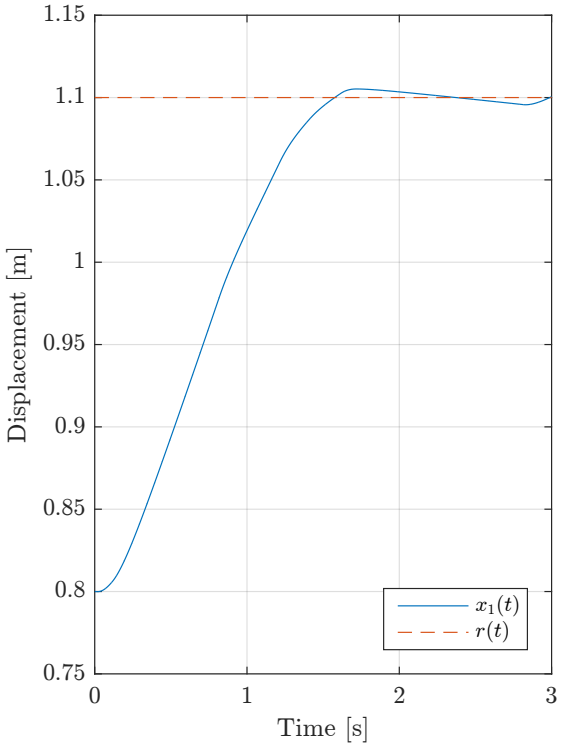


Figure 9.3. NMPC position controller performance with larger input penalty.

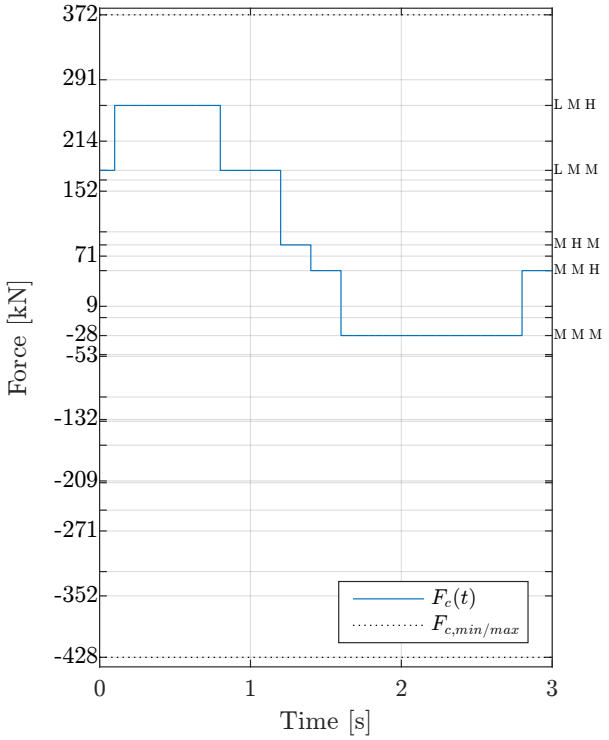


Figure 9.4. Force reference for NMPC position control with larger input penalty.

As seen in the figures the developed NMPC position control is indeed able to penalize the losses associated with force shifts to achieve the desired balance between energy losses and tracking error. As the tracking error and the energy loss are well balanced with the weightings used for the response shown in figure 9.3, this weightings are chosen for the NMPC position controller.

To evaluate how weighting on the tracking error affect the response of the system, the NMPC controller is simulated with different weightings on $Q_1(1,1)$, which is the penalty on tracking error. The second input in the weighting matrix Q_1 is kept constant, in order not to change the penalty on the velocity state. In figure 9.5 simulated results are shown with $Q_1 = \text{diag}[15, 0.1]$, $Q_1 = \text{diag}[2 \cdot 15, 0.1]$, $Q_1 = \text{diag}[0.5 \cdot 15, 0.1]$ and $Q_1 = \text{diag}[0.1 \cdot 15, 0.1]$, while the weightings on input cost and terminal cost are kept at $R_1 = \text{diag}[1, 0.0005]$ and $P_1 = \text{diag}[15, 0.1]$ respectively. It was found, that better results was achieved with no terminal constraints, and resultantly they are omitted.

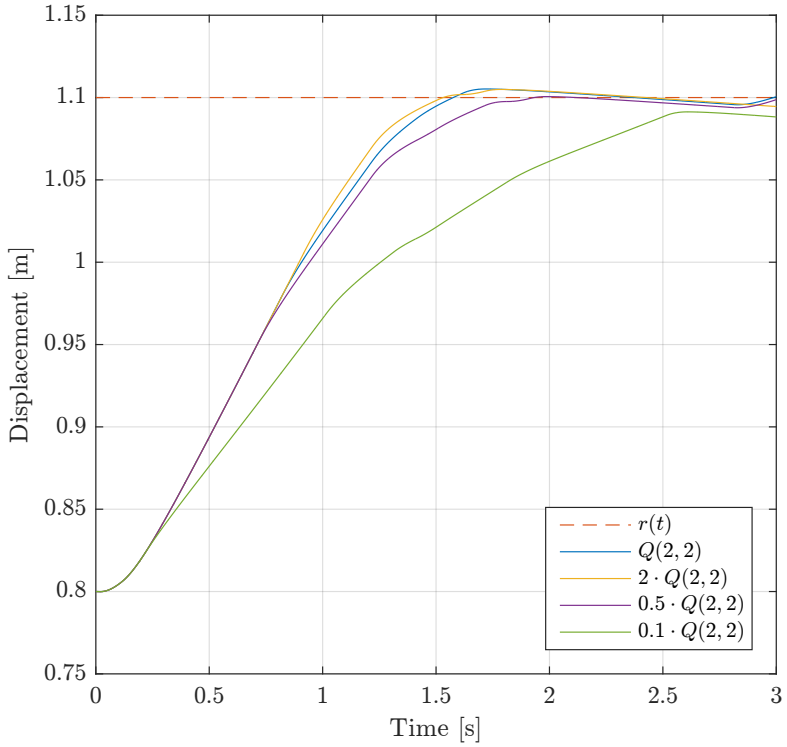


Figure 9.5. NMPC position controller performance for different weightings on tracking error.

As can be seen in the figure an increased weighting on the tracking error result in faster step response. The step response with the weighting $Q_1 = \text{diag}[0.1 \cdot 15, 0.1]$ reach the reference just within the simulation of 3 seconds. The response for the other weightings are similar for the first part of the response, since the tracking error is large. However, as the tracking error decreases the difference between the weightings become clear. It is clear that the position response can be altered by the weighting of the tracking error.

9.4.4 Parameter Tuning for Velocity Control

Similar to the position controller, the NMPC velocity controller is implemented in MATLAB on a simplified model of the system, without transmission line dynamics, in order to test and adjust the optimisation parameters. To evaluate how the controller can balance energy loss and tracking error, a velocity step is made with different weightings. In figure 9.6 and 9.7 a velocity step response and the corresponding force reference are shown with the weightings $Q_2 = P_2 = \text{diag}[0, 2.8]$ and a small input penalty of $R_2 = 10^{-6}\text{diag}[1, 0]$. A similar velocity step response and corresponding force reference is shown in figure 9.8 and 9.9, but with the input penalty $R_2 = \text{diag}[1, 0]$ and the other weightings kept at $Q_2 = P_2 = \text{diag}[0, 2.8]$. For both responses the initial conditions are (1,0) and a constant reference of $r = 0.2$

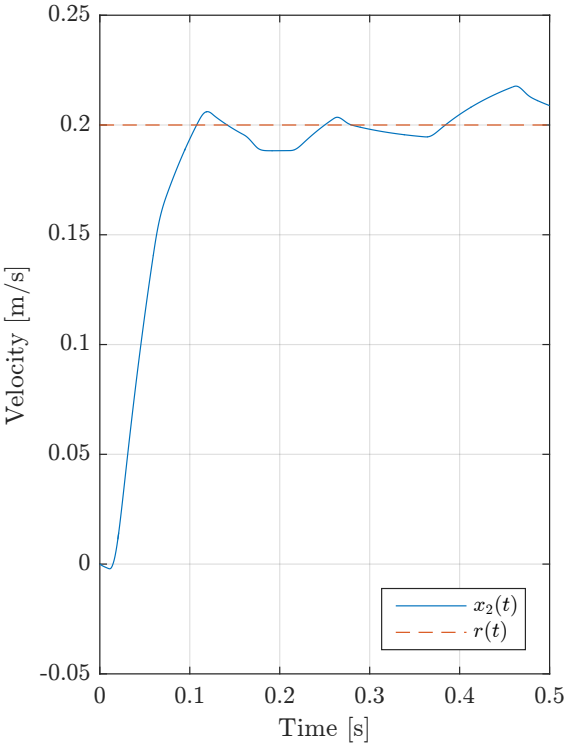


Figure 9.6. NMPC velocity controller performance with small input penalty.

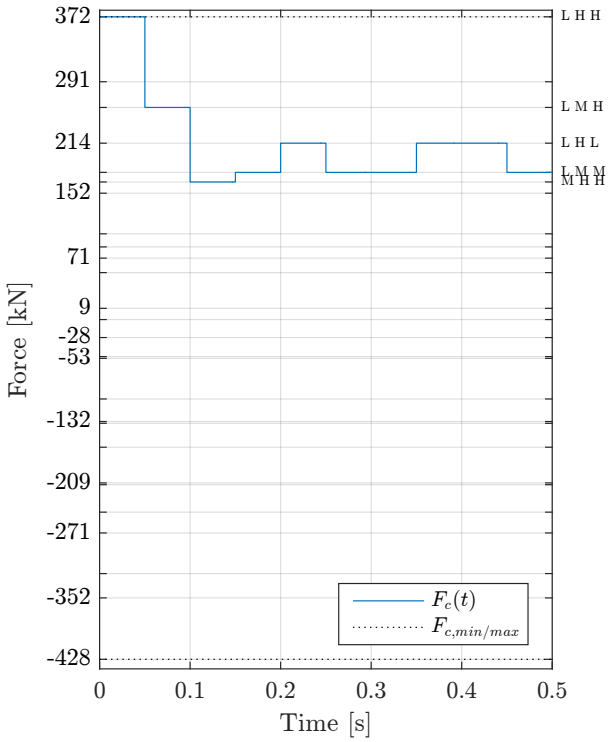


Figure 9.7. Force reference for NMPC velocity control with small input penalty.

As can be seen from figure 9.6 and 9.7 many shifts in force levels are used when the velocity is close to the reference. As the the input penalty is very small, NMPC velocity controller does not consider the losses associated with the force shifts. Assuming the pressure in the chambers are the reference pressure, the energy losses for the velocity step corresponds to approximately 13.6 KJ. In figure 9.9 it is clearly seen that the force shifts are reduced by increasing the input penalty. The corresponding velocity step response shown in figure 9.8 has similar performance as the response with a small input penalty, but the force shifts used to keep the velocity at the reference are reduced. With the same assumption that the pressure in the chambers are the reference pressure, the energy loss for the velocity step shown in figure 9.8 is 4.3 kJ. Thus, the developed NMPC velocity controller is able to balance between the energy losses from shifting and tracking error. Since the weightings used for the velocity response balance the energy losses and tracking error well, these weightings are chosen for the NMPC velocity controller.

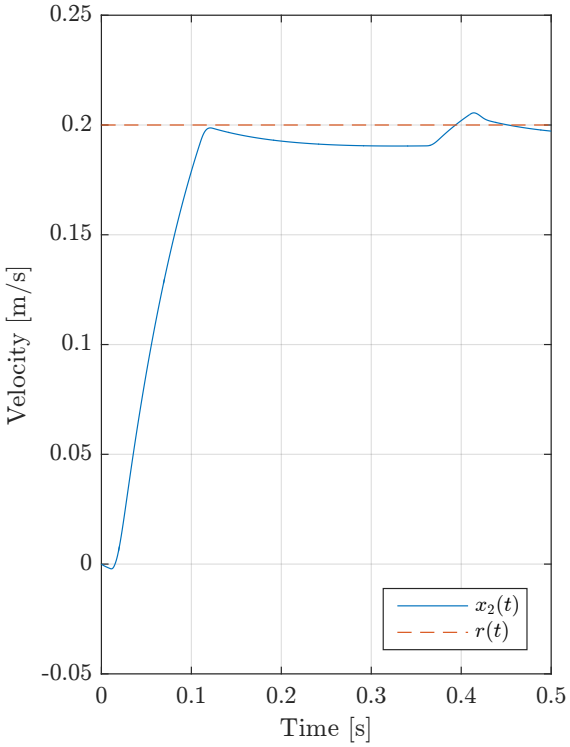


Figure 9.8. NMPC velocity controller performance with larger input penalty.

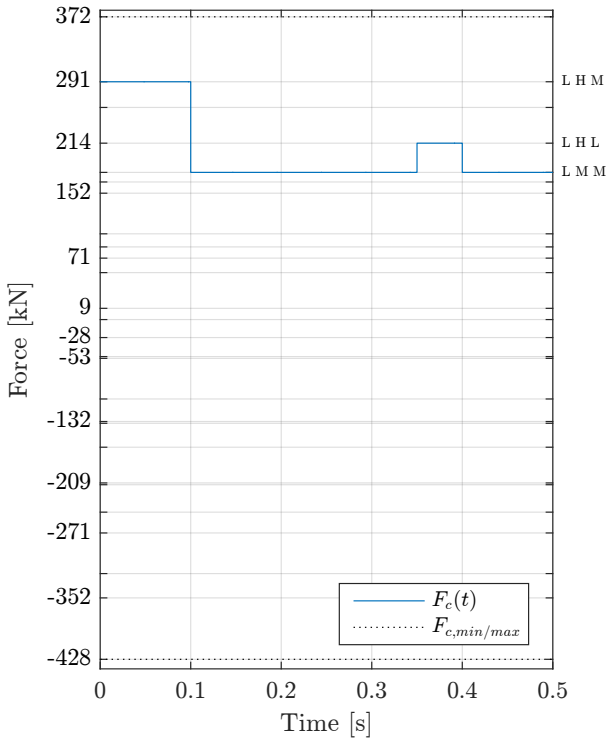


Figure 9.9. Force reference for NMPC velocity control with larger input penalty.

Similar to the NMPC position controller, the response of the NMPC velocity controller is evaluated for different weightings on the tracking error. In figure 9.10 a velocity step response is shown for the weightings $Q_2 = \text{diag}[0, 2.8]$, $Q_2 = \text{diag}[0, 2 \cdot 2.8]$, $Q_2 = \text{diag}[0, 0.5 \cdot 2.8]$ and $Q_2 = \text{diag}[0, 0.1 \cdot 2.8]$, while the weightings on input cost and terminal cost are kept at $R_2 = \text{diag}[1, 0]$ and $P_2 = \text{diag}[0, 2.8]$ respectively. Like for the position control problem, no terminal constraints are set.

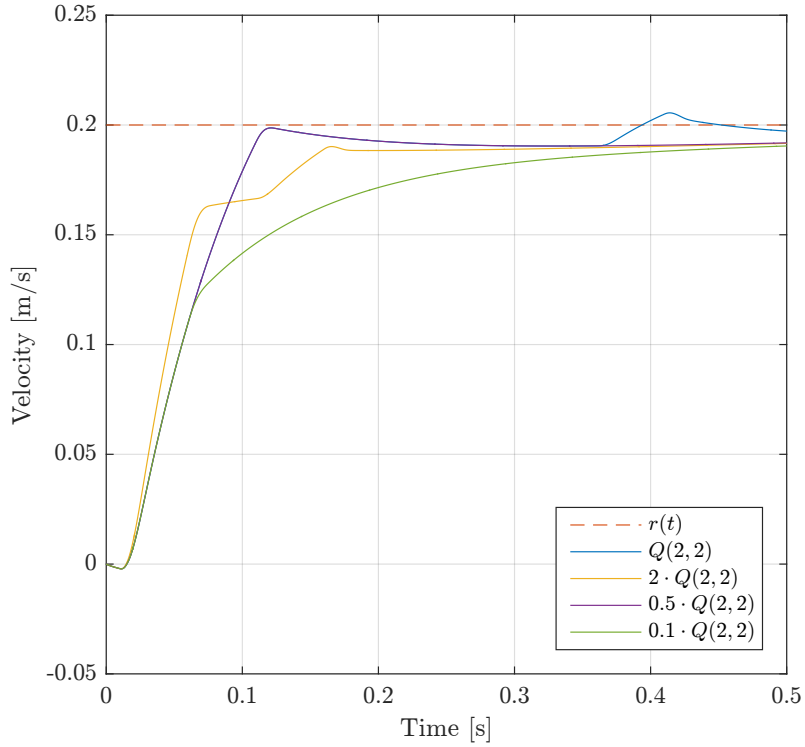


Figure 9.10. NMPC velocity controller performance for different weightings on tracking error.

As seen from the plot the tendency is that an increased weighting on tracking error result in a faster step response. The step response for $Q_2 = \text{diag}[0, 2.8]$ and $Q_2 = \text{diag}[0, 0.5 \cdot 2.8]$ is identical until the last part of the response. However, for the other weightings a clear difference between the weightings is seen. Even though the difference is not as significant as it was for the NMPC position controller, it is seen that the velocity response can be altered by the weighting of the tracking error.

9.5 Results

To evaluate the NMPC controller, it have been implemented in the full order model with the control parameters found in the previous section. In this section results for the NMPC position controller is shown for a step response and for load holding. Results for the NMPC velocity controller is evaluated with a step response and a sine wave reference trajectory.

Common for all results is that by using the full model larger pressure oscillations occurs, due of the introduction of transmission line dynamics. These oscillations are also seen in the force output.

9.5.1 Position Control

A step in position with the NMPC position controller used on the full order model is seen in figure 9.11. The displacement in the top left corner shows similar response as seen for the simplified model used for tuning the control parameters. The controller is able to reach the reference in approximately 2 seconds using 10 force shifts. The RMS-error for this step response is, e_{RMS}=125.5mm. Mainly switching in low volume chambers are preferred by the NMPC, as this reduces switching losses.

Resultantly, only 5 force levels are utilised. The switching loss for the step is, E_β=9.976kJ. This switching loss is calculated as in the input cost function in section 9.2.

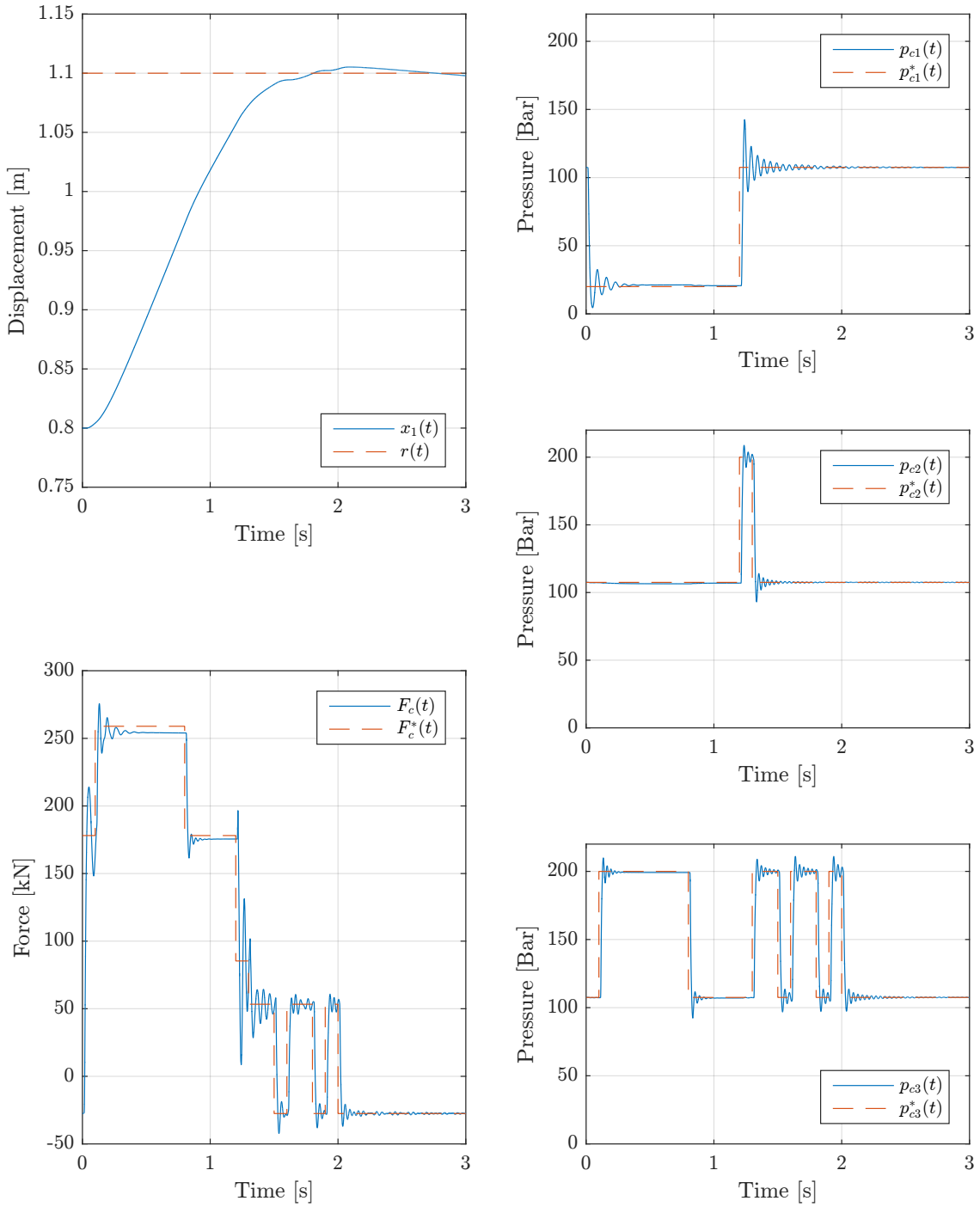


Figure 9.11. Step response with NMPC position control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

The load holding performance is shown in figure 9.12. A slight error is introduced in order to see the behaviour. It is seen, that the NMPC has oscillating behaviour, but the switching losses are low as only the low volume chamber is used for correcting position close to zero error. The RMS-error for load holding is e_{RMS}=4.3mm.The switching loss is E_β=5.905kJ.

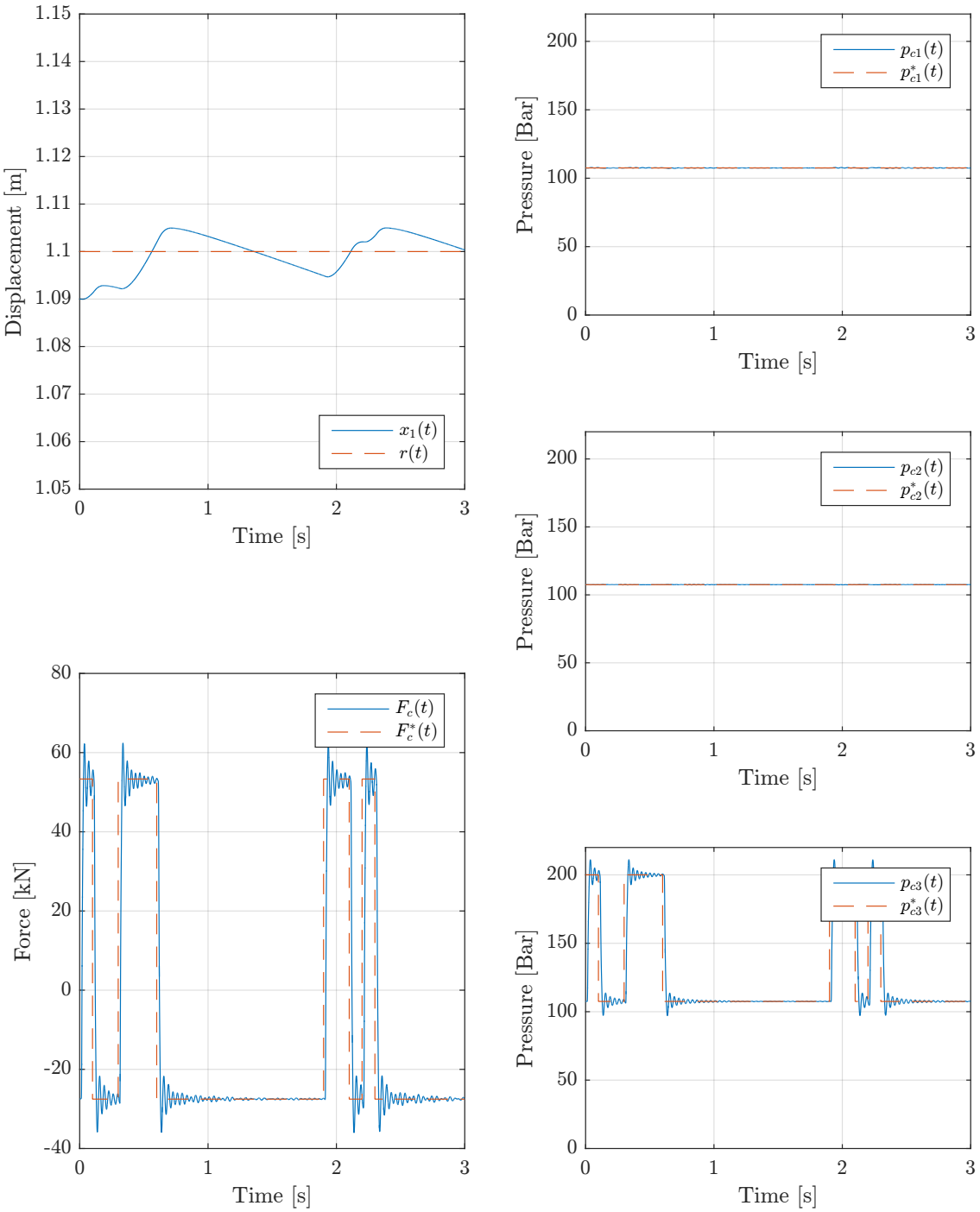


Figure 9.12. Step response with NMPC velocity control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

9.5.2 Velocity Control

A step in velocity with the NMPC velocity controller used on the full order model is seen in figure 9.13. The controller reaches the reference in just over 0.1s with only one force step. After this two more steps are made to maintain the velocity. The RMS-error is, e_{RMS}=61.3mm/s and the switching loss is, E_β=4.448kJ.

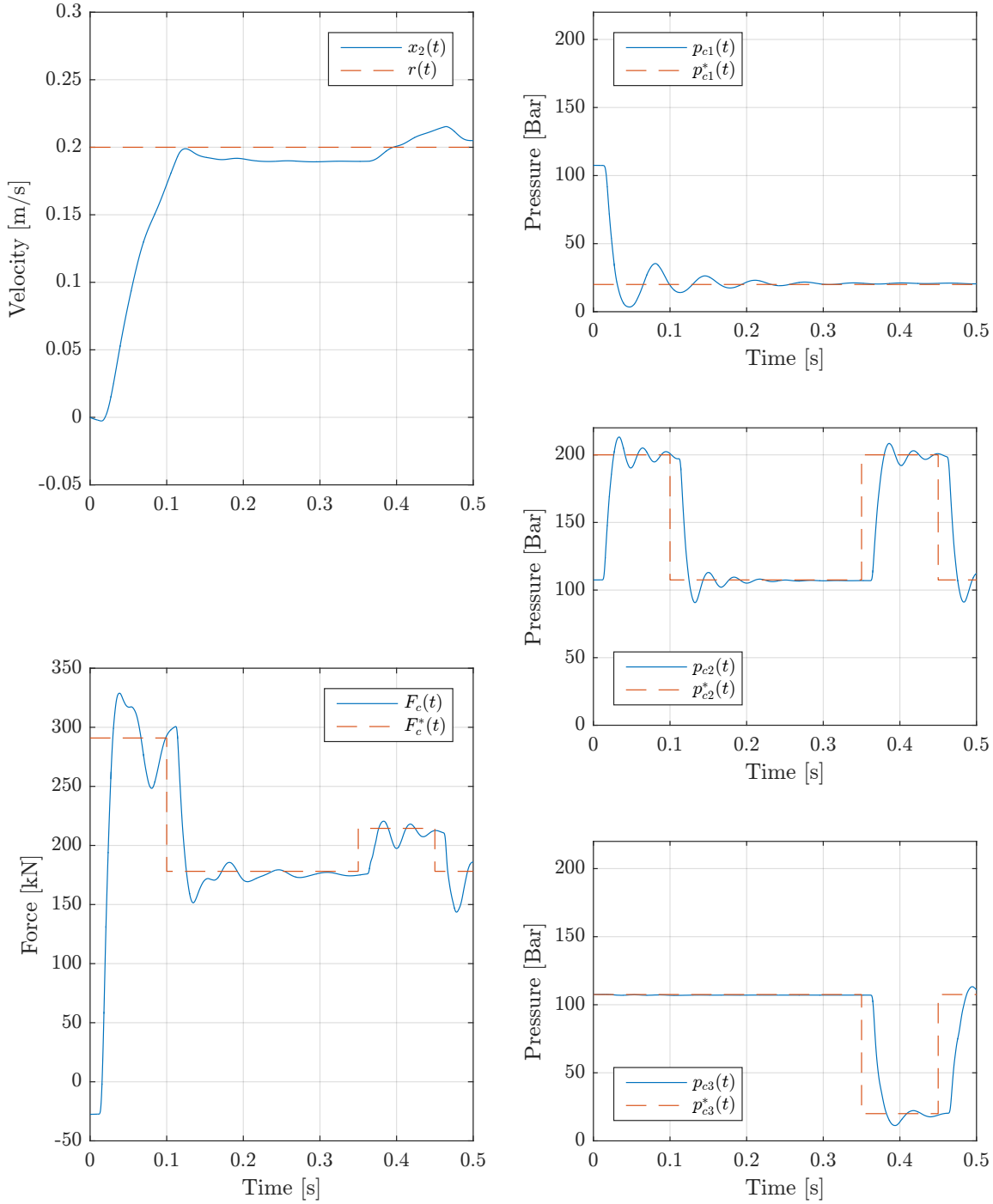


Figure 9.13. Step response with NMPC velocity control. Top left corner is velocity, bottom left is cylinder force and the three plots to the right are chamber pressures.

In figure 9.14 the NMPC is set to follow a sine trajectory. Only a small phase lag is seen and NMPC tracks the velocity well. A high amount of switching on chamber 1 is seen around 2 seconds, yielding large oscillations in force output. If this behaviour is undesirable, the cost function can be adjusted accordingly at the cost of tracking performance. The RMS-error is e_{RMS}=27.3mm/s and the switching cost is E_β=93.68kJ.

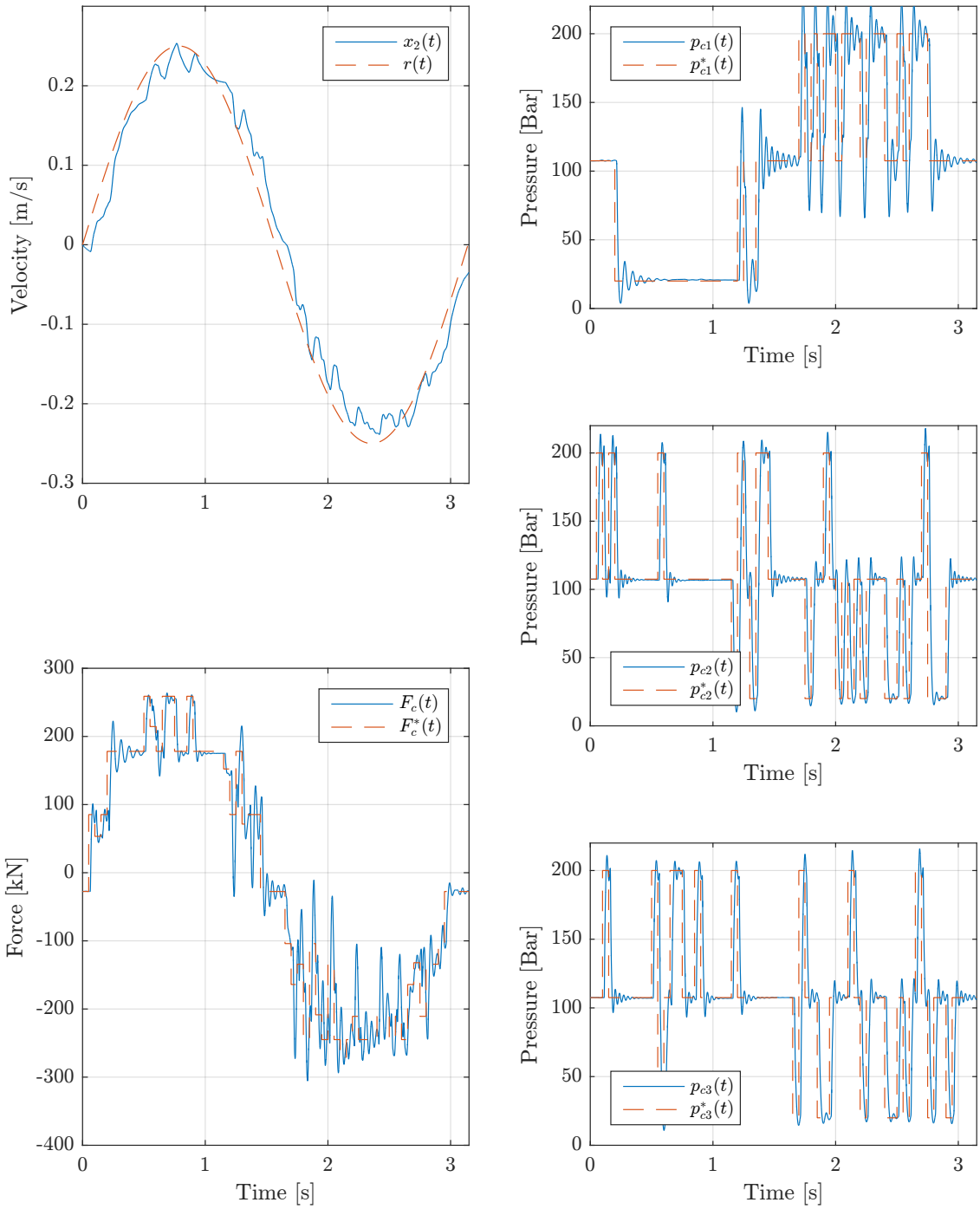


Figure 9.14. Sine wave reference trajectory with NMPC velocity control. Top left corner is velocity, bottom left is cylinder force and the three plots to the right are chamber pressures.

9.6 Chapter Summary

Concluding this NMPC design chapter, an NMPC problem has been formulated based on a discrete time reduced order model of the application case in section 7.2. This included a cost function that penalises tracking error as well as losses associated with switching the DDC chambers between discrete pressure levels. The Euler time step size was chosen based on system settling time as well as the pressure dynamics of the DDC.

NMPC has been shown to reduce the losses associated with switching while minimising the error on both position and velocity by weighting the cost function appropriately. However, it has also been shown, that it is not possible to achieve these results in a real-time implementation as the computational effort required is too large for today's standards. Hence, a more efficient method must be investigated on how to achieve similar results in a real-time implementation.

The next chapter concerns the development of an explicit implementation of the NMPC control presented in this chapter.

CHAPTER 10

DESIGN OF EXPLICIT MODEL PREDICTIVE CONTROL

As previous mentioned, a disadvantage of NMPC is the large computational power required to solve the optimisation problem in real time. One way to address this problem is by eMPC. An approximate multi-parametric Nonlinear Integer Programming (mp-NIP) approach can be used to design an eMPC controller for constrained nonlinear systems with quantized control inputs [19]. The idea of the approach is to construct a piecewise constant approximation to the optimal solution of the mp-NIP problem, such as the one summarised in section 9.4. In this chapter the eMPC approach presented in [19] is first described followed by a step-by-step procedure of the approach. The procedures used in the individual steps will thereafter be described in details. By the end of the chapter the parameters used for the implemented eMPC controller is presented together with an evaluation of the solution. Furthermore, convergence tolerances are discussed as well as steps taken to reduce convergence times.

10.1 Algorithm

The main idea of eMPC for systems with quantized inputs is to construct a hyper-rectangle X that cover the whole partition of the state space. The partition of the state space is bounded by the variable limits. Hyper-rectangles covering X is then constructed. For each hyper-rectangle covering X, a set of interior points are computed an denoted $W_0 = \{w_0, w_1, \ldots, w_n\}$, where W_0 is the set of points associated to the hyper-rectangle X_0. W_0 consists of the center point of the hyper-rectangle X_0, and the vertices and facets center of one or more hyper-rectangles contained in the interior of X_0. The procedure to generate the set of points associated to a hyper-rectangle will be explained later.

A close-to-global solution $\mathbf{U}^*(w_i)$ at a point $w_i \in W_0$ is computed using the routine *glcFast* of the TOMLAB optimisation environment in MATLAB, which is based on the DIRECT optimisation algorithm presented in appendix B. Based on the close-to-global solution at all points $w_i \in W_0$, a local constant approximation $\hat{\mathbf{U}}_\mathbf{0}(\mathbf{x}) = \mathbf{K_0} = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_N]$ to the optimal solution $\mathbf{U}^*(\mathbf{x})$, valid in the whole hyper-rectangle X_0, is determined. The procedure to determine $\hat{\mathbf{U}}_\mathbf{0}(\mathbf{x})$ will be presented in section 10.2.

An estimate of the error bound between the cost function with the local constant approximation $\hat{\mathbf{U}}_\mathbf{0}(\mathbf{x})$ and cost function with the optimal solution $\mathbf{U}^*(\mathbf{x})$ for the hyper-rectangle X_0 is denoted $\hat{\epsilon}_0$.

A comparison between $\hat{\epsilon}_0$ and a maximum approximation error $\bar{\epsilon}$ set by the user determines whether the local constant approximation is sufficiently close to the optimal solution. The procedure to determine $\hat{\epsilon}_0$ is presented later.

The algorithm used to compute the explicit solution for NMPC with quantized control input can be expressed as the following step-be-step algorithm. A flow diagram of the algorithm can be seen in figure 10.1.

Step 1. Initialisation of the partition of the whole hyper-rectangle, i.e. $\Pi = \{X\}$. The hyper-rectangle X is marked unexplored.

Step 2. Any unexplored hyper-rectangle $X_0 \in \Pi$ is selected. If no unexplored hyper-rectangle exists the algorithm is terminated.

Step 3. A set of points $W_0 = \{w_1, w_2, ...w_n\}$ associated to X_0 is generated.

Step 4. A solution to the optimisation problem for $\mathbf{x}$ fixed to each of the points in the set W_0 is generated, using the routine *glcFast* of TOMLAB optimization environment. If the optimisation problem has a feasible solution at all these points, the algorithm goes to step 7. Otherwise it continue to step 5.

Step 5. The size of the hyper-rectangle X_0 is computed by calculating the area of the hyper-rectangle. If the size is smaller than a given tolerance ϵ_v, X_0 is marked infeasible and explored and the algorithm goes to step 2. Otherwise it continues to step 6.

Step 6. X_0 is split into 4 new hyper-rectangles X_1, X_2, X_3 and X_4, by a hyperplane through the center point and orthogonal to each axis. The 4 new hyper-rectangles is marked unexplored and X_0 is removed from Π. The algorithm goes to step 2.

Step 7. A constant function $\hat{\mathbf{U}}_\mathbf{0}$ to be used in the hyper-rectangle X_0 is computed. If a feasible solution is found the algorithm continues to step 8. Otherwise X_0 is split into two new hyper-rectangles X_1 and X_2, by a hyperplane through the center point and orthogonal to an arbitrary axis. X_1 and X_2 are marked unexplored and X_0 is removed from Π and the algorithm goes to step 2.

Step 8. An estimate of the error bound $\hat{\epsilon}_0$ is computed. If $\hat{\epsilon}_0 \leq \bar{\epsilon}$, X_0 is marked explored and feasible, and the algorithm goes to step 2. Otherwise X_0 is split into two new hyper-rectangles X_1 and X_2, by a hyperplane through the center point and orthogonal to the axis with the largest error gradient. X_1 and X_2 are marked unexplored and X_0 is removed from Π and the algorithm continue to step 2.

The algorithm continues until the hyper-rectangle covering the whole partition of the state space X, is divided into smaller hyper-rectangles, that all have a feasible constant solution or contain infeasible points and a size smaller than the given tolerance.

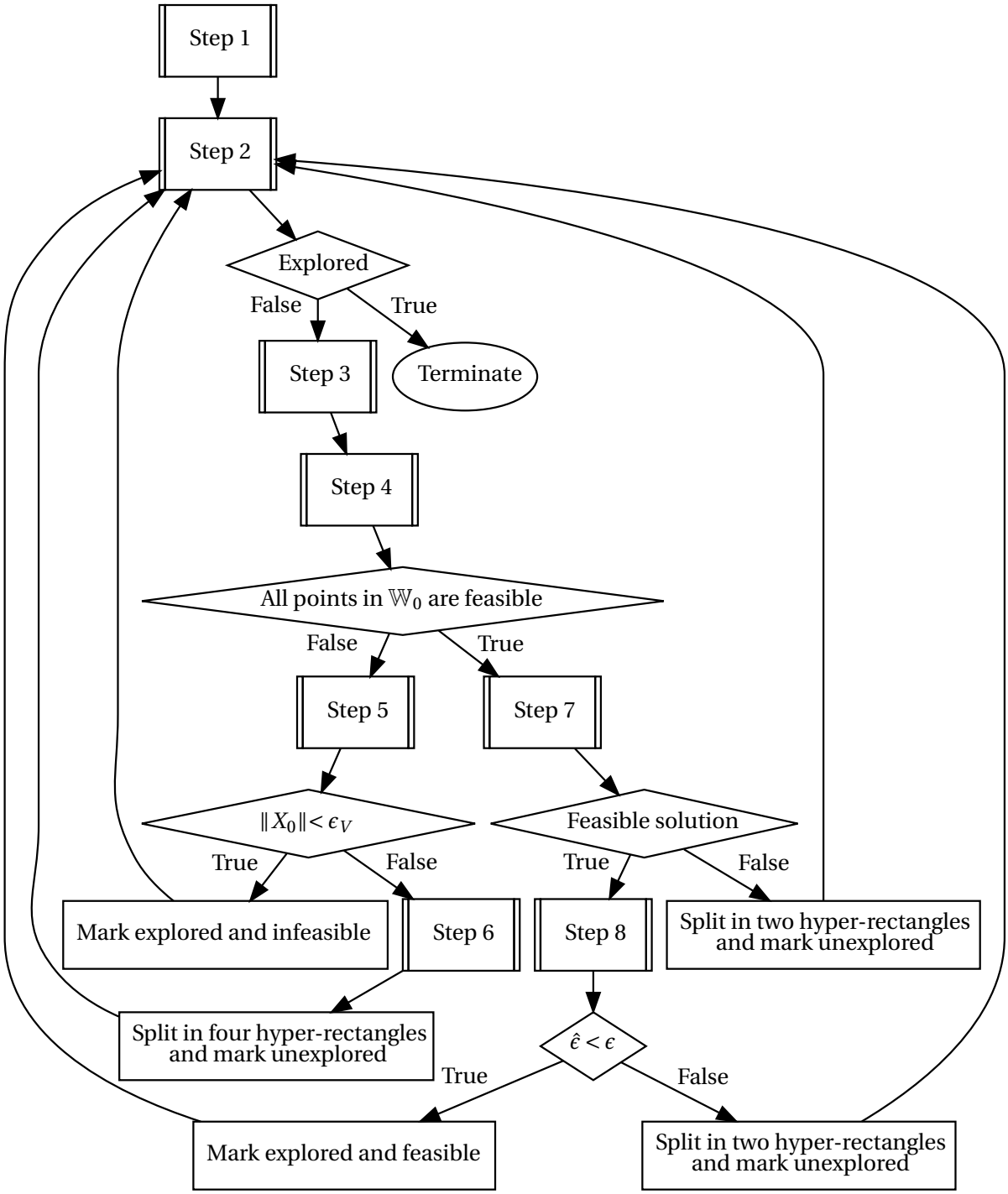


Figure 10.1. Flow diagram of the algorithm used to compute the explicit solution for NMPC with quantized control inputs.

The algorithm is based on the close-to-global solution to a set of points W_0 in each hyper-rectangle. The generated points used in the implemented algorithm for a two dimensional hyper-rectangle is shown in figure 10.2. The set of points W_0 corresponds to the center point of hyper-rectangle X_0 denoted w_0 in the figure and the vertices $(w_9, w_{11}, w_{13}, w_{15})$ and facets center $(w_{10}, w_{12}, w_{16}, w_{14})$ of hyper-rectangle X_0. Furthermore, the vertices (w_1, w_3, w_5, w_7) and facets center (w_2, w_4, w_6, w_8) of the hyper-rectangle X_0^1 contained in the interior of X_0 is used. More hyper-rectangles in the

interior of X_0 could be used to generate more points, which would result in a higher resolution and an increased number of points that should be optimised, which would increase the computation time of the algorithm.

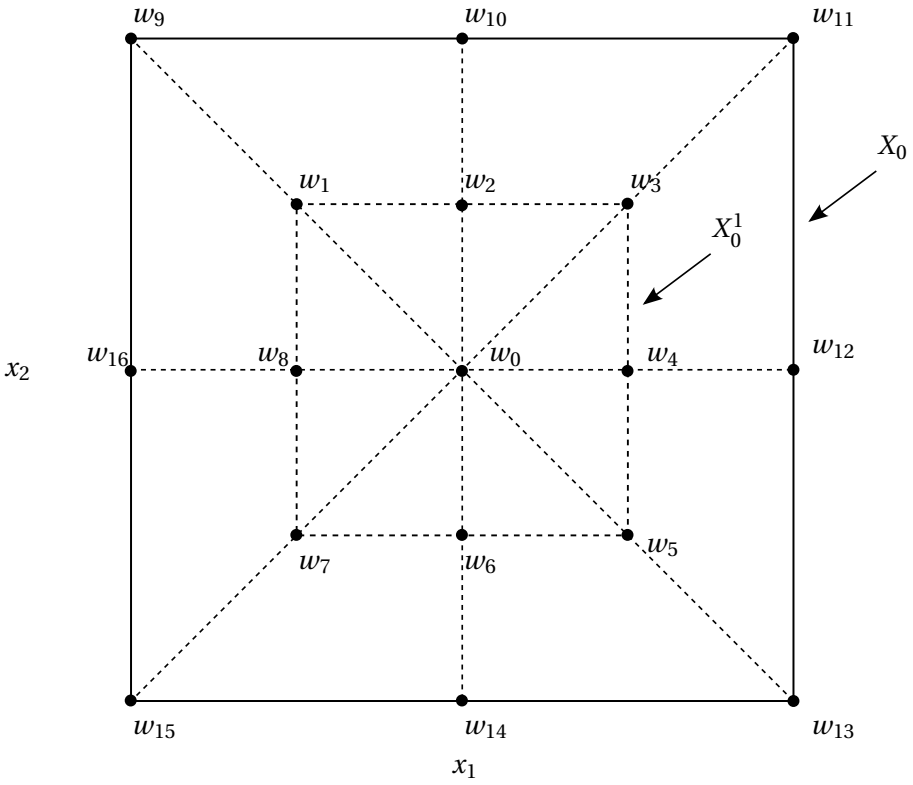


Figure 10.2. Illustration of how set of points for which a optimal solution is determined.

The hyper-rectangle X_0^1 has the same center point as X_0 and half the length of the sides. If one of the generated points has an infeasible solution the hyper-rectangle is divided into the 4 hyper-rectangles X_1, X_2, X_3 and X_4 as illustrated in figure 10.3. The infeasible points in the figure are w_1, w_9, w_{10} and w_{11}. The hyper-rectangle X_0 is divided by a hyperplane through the center point and orthogonal to each axis. Other splitting strategies could be used, in order to faster divide the whole hyper-rectangle into feasible and infeasible hyper-rectangles and thereby decrease the computation time of the algorithm. In [5] a heuristic splitting rule is used to split the hyper-rectangle into feasible and infeasible hyper-rectangles. However, the heuristic splitting rule is not used in the implemented algorithm.

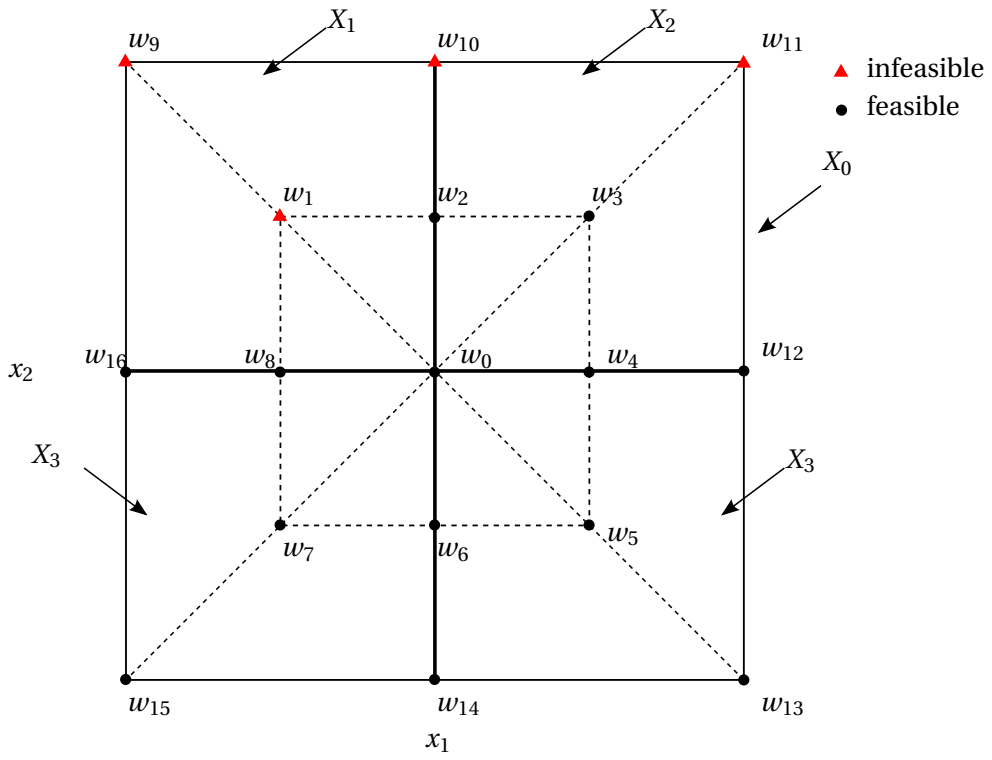


Figure 10.3. Illustration of how the hyper-rectangle is divided if it contains at least one infeasible point.

If $\hat{\epsilon}_0 \geq \bar{\epsilon}$, X_0, the hyper-rectangle is split into two new hyper-rectangles X_1 and X_2. X_0 is split by a hyperplane through the center point and orthogonal to the axis with the largest error gradient. By splitting orthogonal to the axis with the largest error gradient the estimate of the error is more likely to be smaller, because the objective function values are more uniform in the new hyper-rectangles. This and other step done to reduce computational time is further described in section 10.4.1.

10.2 Local Constant Approximation Problem

As previous described the close-to-global solution $\mathbf{U}^*(w_i)$ at each point $w_i \in W_0$ is computed using the routine *glcFast* of the TOMLAB optimisation environment in MATLAB. The local constant approximation $\hat{\mathbf{U}}_\mathbf{0}(\mathbf{x}) = \mathbf{K_0}$ is then computed by solving the following NIP

$$\min_{K_0 \in U^B} \sum_{i=0}^{n} \left(J(\mathbf{K_0}, w_i) - V^*(w_i) \right) \tag{10.1}$$

subject to state equations 9.2, 9.3, $\mathbf{x}(t|t) = \mathbf{x}(t)$ and

$$\begin{aligned} & u_i(t+k|t) \in \{1,2,3\} \qquad k = 0,...,N-1 \quad , i = 1,2,3 \\ & \left.\begin{array}{rll} 0 & \leq x_1(t+k|t) & \leq L_s \\ -0.5 & \leq x_2(t+k|t) & \leq 0.5 \end{array}\right\} \qquad k = 1,..,N-1 \\ & \mathbf{x}(t+N|t) \in \mathbb{X}_{Tj} \end{aligned}$$

The NIP is solved using the *glcFast* routine. If a feasible solution of $\hat{\mathbf{U}}_\mathbf{0}(\mathbf{x}) = \mathbf{K_0}$ associated to the hyper-rectangle X_0 is found, the cost approximation error in X_0 is

$$\epsilon(\mathbf{x}) = \hat{V}(\mathbf{x}) - V^*(\mathbf{x}) \leq \epsilon_0 \quad , \quad \mathbf{x} \in X_0 \tag{10.2}$$

where $\hat{V}(\mathbf{x}) = J\left(\hat{\mathbf{U}}_{\mathbf{0}}(\mathbf{x}), \mathbf{x}\right)$ is the sub-optimal cost function and $V^*(\mathbf{x})$ is the cost function corresponding to the close-to-global solution $V^*(\mathbf{x}) = J(\mathbf{U}^*(\mathbf{x}), \mathbf{x})$. An estimate of the maximal approximation error ϵ_0 can be calculated as

$$\hat{\epsilon}_0 = \max_{i \in \{0,1,\ldots,n\}} \left(\hat{V}(w_i) - V^*(w_i)\right) \tag{10.3}$$

10.3 State Space Partitioning

The initial state space partitioning of **Step 1** defines the space on which the controller (and system) is expected to operate. Hence, it is natural to bound this by physically motivated limitations, such as stroke length and a maximum desired velocity. However, a large state space partition lead to a large optimisation problem to be solved. In order to reduce the optimisation problem, the state space is modified to be described by the error, such that

$$e(t) = \begin{cases} r(t) - y(t) & \text{if} \quad e_{min} < r(t) - y(t) < e_{max} \\ e_{min} & \text{if} \quad r(t) - y(t) < e_{min} \\ e_{max} & \text{if} \quad e_{max} < r(t) - y(t) \end{cases} \tag{10.4}$$

Since $y(t)$ (either x_1 or x_2) are needed for the state model, cost and constraint functions, it is calculated as

$$y(t) = r(t) - e(t) \tag{10.5}$$

The initial state space partition Π_j is then a hyper-rectangle in three dimensions. For position control the initial state space is then defined by

$$\Pi_1 = \left[e_{1,min}, e_{1,max}\right] \times \left[x_{2,min}, x_{2,max}\right] \times \left[r_{min}, r_{max}\right] \tag{10.6}$$

while for velocity control

$$\Pi_2 = \left[x_{1,min}, x_{1,max}\right] \times \left[e_{2,min}, e_{2,max}\right] \times \left[r_{min}, r_{max}\right] \tag{10.7}$$

However, the cost function includes a derivative of the input. Thus, an initial condition of the input must be defined in each point. These variables are discrete, hence intermediate points are not feasible and the algorithm described above does not apply. Instead, the state space Π_j is expanded in a discrete fourth dimension. Since there are 27 input combinations, the state space partition is expanded to 27 three dimensional hyper-rectangle and thus the algorithm must run for each of these hyper-rectangle.

10.4 Convergence Parameters

The decision of whether a local explicit approximate solution is feasible in **Step 8** relies on the definition of $\bar{\epsilon}$. Decreasing $\bar{\epsilon}$, the convergence time is increased and a smaller approximation error is achieved, while increasing $\bar{\epsilon}$ decreases convergence time and increases approximation error. Thus, defining $\bar{\epsilon}$ relies on a balance between approximation error and convergence time. The the cost function optimum value V^* varies as a function of initial condition. If the initial condition is close to the reference, the cost function can be expected to be smaller than if the initial condition is far from the reference. Hence, the state space partitioning defined in section 10.3 makes $V^*(\omega_i)$ more uniform in X_0 by limiting the maximum error. Still, to ensure a finite convergence time, $\bar{\epsilon}$ is defined to include

both an absolute approximation error tolerance, ϵ_a, and a relative approximation error tolerance, ϵ_r. $\bar{\epsilon}$ is then defined by

$$\bar{\epsilon} = \max\left(\epsilon_a, \epsilon_r \min V^*(\omega_i)\right) \tag{10.8}$$

This ensures that $\bar{\epsilon}$ is defined by the minimum V^* within X_0. If $V^*(\omega_i)$ is not sufficiently uniform within X_0, X_0 should be split.

10.4.1 Reducing Computational Time

Convergence time is ultimately a parameter on which an optimisation algorithm is evaluated. Hence, any steps that can be done to reduce convergence time should be explored. As mentioned in section 10.1, the authors of [29] note that a heuristic splitting rule is beneficial when splitting a set of points where one or more are infeasible. However, for this application, infeasible points only exists in near the limits of the state space partition and are few in numbers. Thus, the time developing code applying heuristics is not well spent.
However, other steps where taking to reduce the computational time of the algorithm.

Re-appearing Points

It was found that points in W_0 generated in **Step 3** often have already been optimised in earlier iterations. Depending on the amount of sub hyper-rectangles, the number of re-appearing points constituted between 10% and 70% of the optimisation problem in **Step 4**. Since **Step 4** is the most time consuming step in the algorithm, code was developed in order to recognize when points in W_0 had previously been optimised and then exclude those points from the optimisation in **Step 4**. On average, this reduced the computational time in **Step 4** by 60%.
The point recognising code was added as a part of **Step 4** and its MATLAB implementation can be seen in appendix D.2.

Approximation Error Gradient

In the description of the algorithm in [19] it is stated that in **Step 8** the splitting of the hyper-rectangle is done through an arbitrary axis. However, in [29] the authors note, that the hyper-plane splitting the hyper-rectangle is selected such that the change of approximation error is maximal across the hyper-plane. This is done to minimise the approximation error in each new hyper-rectangle. This has been implemented in **Step 8**, such that the approximation error gradient is estimated for each one axis-orthogonal hyperplane to split the partition.
The implementation can be seen in appendix D.6 as a part of **Step 8**.

In **Step 7**, if no feasible solution is found, the largest error gradient condition cannot be used for selecting a splitting hyper-plane. It is likely that infeasibility of K_0 is due to too large parameter variations in W_0. Hence, W_0 is split with a hyperplane orthogonal to the axis in which W_0 has the longest facet.

Adaptive Tolerances

In a similar manner as **Step 5**, a volume tolerance is used to establish a soft limit on the size of the hyper-rectangles in **Step 8**. If the volume of the hyper-rectangle is below a given threshold, the absolute and relative tolerances are increased by a constant factor. Beside having implications on the computational time, this also has a practical aspect. The volume threshold defines an average

soft minimum on the length of the faces of the hyper-rectangle, which can be related to the quality of state measurement. Hence, if a maximum measurement accuracy is e.g. ±0.5 mm, ±0.5 mm/s, a minimum volume threshold can be set to 1mm·1mm/s. This implementation is not a hard direct limit on state, as the hyper-rectangle is not necessarily a square. A more direct limit on the size of state variations of the hyper-rectangles can be implemented in a similar manner, but as this thesis is mainly focused on simulating results, this was not implemented.

10.4.2 Control implementation

As described in section 10.3 an explicit solution is computed for each initial combination of the input, which due to the 27 input combinations result in 27 hyper-rectangles each covering the whole state space Π_j. Each of the 27 hyper-rectangles that cover the whole state space are denoted Π^d where $d = 1, 2..., 27$. Each Π^d consist of a number of smaller hyper-rectangles that either have an associated feasible control function $\mathbf{K}_0$ or no feasible solution, the total number of hyper-rectangles in Π^d is denoted n_z^d. In the following i will be used to denote an arbitrary hyper-rectangle in Π^d. Each hyper-rectangle is defined by two vectors, $\hat{z}_i^d$ and $\check{z}_i^d$, which contains the maximum and minimum state values respectively. We denote the set of maximum and minimum points $z_i^d = [\hat{z}_i^d, \check{z}_i^d]^T$. The control value associated with z_i^d is $\mathbf{K_{0i}^d} = [\mathbf{k}_{1i}^d, \mathbf{k}_{2i}^d, \ldots, \mathbf{k}_{Ni}^d]$. With the measured state at time t equal to $\mathbf{x}(t|t)$, the control output at time t is selected such that

$$\mathbf{u}(t) = \mathbf{k}_{1i}^d \in \mathbf{K_{0i}^d} \tag{10.9}$$

where i is chosen such that, e.g for position control

$$\begin{array}{lll} \check{z}_{1i}^d & \leq r(t) - x_1(t|t) & \leq \hat{z}_{1i}^d \\ \check{z}_{2i}^d & \leq x_2(t|t) & \leq \hat{z}_{2i}^d \end{array} \tag{10.10}$$

and d is chosen based on the previous input. The computational time of the above is in the order of milliseconds for $n_z \sim 1000$.

10.5 Problem Summary

The above algorithm requires a large amount of computational time to complete. Thus, it is not feasible to test and tune parameters as was done for the NMPC problem. The approach must be to tune the NMPC problem before converting the solution to an explicit form. Hence, optimisation problem is the same as the one summarised in 9.4 and the values for Q_j, R_j, S_j, N, T_s are readily tuned and are stated here
For position control, the horizon is chosen as $N = 15$ and time step size $T_s = 0.1$.

$$Q_1 = P_1 = \text{diag}[15, 0.1], \quad N_1 = \text{diag}[1, 0.0005] \tag{10.11}$$

For velocity control, $N = 10$ and time step size $T_s = 0.05$.

$$Q_2 = P_2 = \text{diag}[0, 2.8], \quad N_2 = \text{diag}[1, 0] \tag{10.12}$$

Both problems have no terminal constraints. Like for the NMPC problem, the optimisation problems in **Step 4** and **Step 7** are solved with 4000 function evaluations.

The MATLAB implementation of the algorithm described in section 10.1 is seen in appendix D.

10.5.1 Three Dimensional State Space

As mentioned the initial state space is naturally bounded by the physically limits of the system. As shown in 10.3 this was limited by the state space partitioning to reduce the optimisation problem. Ideally the explicit solution to the initial state space Π_j should cover as much of the physical limits as possible within the computational capacity.

In figure 10.4 the explicit solution is illustrated for the three continuous dimensions in one of the initial discrete force levels. The state space seen is; $\Pi_1 = [-0.2, 0.2] \times [-0.4, 0.4] \times [0.75, 1.25]$. The figure shows the hyper-rectangles for the state space when 82.9% of the volume have been explored. At this time 5800 regions have been found, both explored and unexplored. In order to speed up the computation, the algorithm for this computation was made with 800 function evaluations in the numerical solver. Even with a limited state space and low amount of function evaluations, it took more than 39 hours to compute the results for 82.9% for one force level. Thus, with limited time and computational capacity for this thesis it is necessary to investigate how to reduce the problem size.

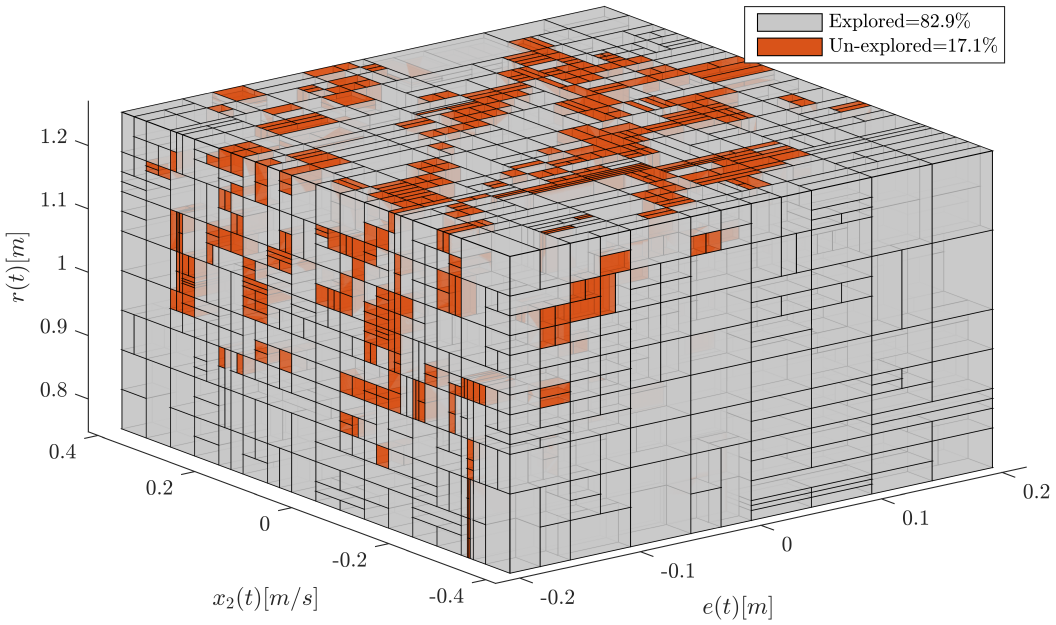


Figure 10.4. Illustration of explicit solution for three continuous dimensions for one set of discrete pressure levels (all high pressure). 5800 regions found after 82.9% of the volume was marked as explored.

10.6 Reducing Problem Size

The size of the eMPC problem is mainly described by the dimension of the problem. The problem above has three continuous dimensions as well as a fourth discrete dimension. Like other optimisation problem, this problem suffers from the "curse of dimensionality", where problem size grows exponentially with the amount of problem dimensions. Due to low computational power available, the problem size has been reduced in order to achieve results before project deadline.

Firstly, one dimension has been made constant. This reduces the amount of continuous dimension from three to two. Furthermore, the maximum error size defined in section 10.3 has been reduced as

well as the maximum allowed state values.

The continuous state space is then described by

$$\Pi_1 = \left[e_{min}, e_{max}\right] \times \left[x_{2,min}, x_{2,max}\right] \tag{10.13}$$

where $e_{max} = -e_{min} = 0.3$ [m], $x_{2,max} = -x_{2,min} = 0.4$ [m/s] with constant reference $r = 1.1$ [m]. The discrete dimension is not reduced. The tolerances ϵ_a and ϵ_r are chosen based on calculated V^* for the initial partition. ϵ_a is chosen as half of the minimum value of V^*, such that $\epsilon_a = 0.01$ and the relative tolerance is chosen as $\epsilon_r = 0.05$.

The continuous state space for velocity control is described by

$$\Pi_2 = \left[e_{min}, e_{max}\right] \times \left[r_{min}, r_{max}\right] \tag{10.14}$$

where $e_{max} = -e_{min} = 0.25$ [m/s], $r_{max} = -r_{min} = 0.25$ [m/s] with constant position $x_1 = 1$ [m]. Hence, the position dependency for the switching losses are neglected as well as the position dependent mass and gravitational force. The discrete dimension is not reduced.

The absolute tolerance is chosen as $\epsilon_a = 0.02$ and the relative tolerance is chosen as $\epsilon_r = 0.1$ for the velocity control problem. No sub hyper-rectangles are created in **Step 3**, and hence a hyper-rectangle contains 9 points in two dimensions. This is considered reasonable, as the solution to the problem is assumed sufficiently smooth, i.e no internal infeasible points or large differences in cost function value.

10.6.1 Two Dimensional State Space for Position Control

An explicit solution have been found for the reduced position problem described above. The solution space consists of 27 solution planes for each of the 27 force levels (illustrated on figure 3.4 on page 12). Two examples of these planes are shown in figure 10.5 and 10.6. The solutions shown consists of 243 and 264 hyper-rectangles for the initial force levels of 14 and 3 respectively. The number at each hyper-rectangle shows the first force level in the horizon of the approximated solution. The colors of each rectangle corresponds to a force, shown on the colorbar to the right.

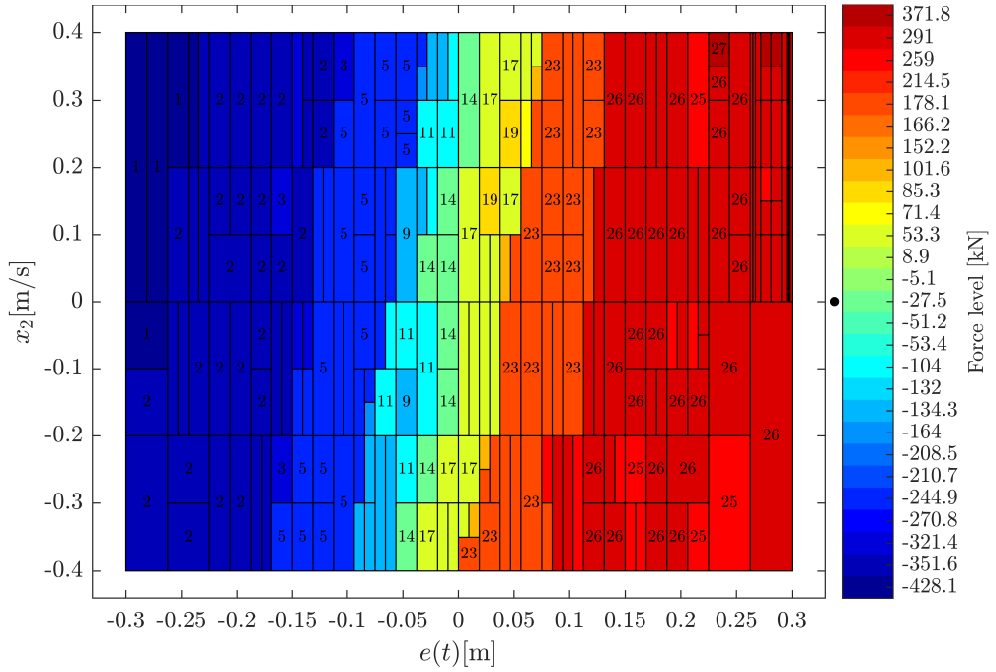


Figure 10.5. Two dimensional state space for eMPC position control. Initial force level is 14, corresponding to MMM and marked with a bullet next to the colorbar. Colors note the first force level in the horizon of the approximated solution for each of the 243 hyper-rectangles.

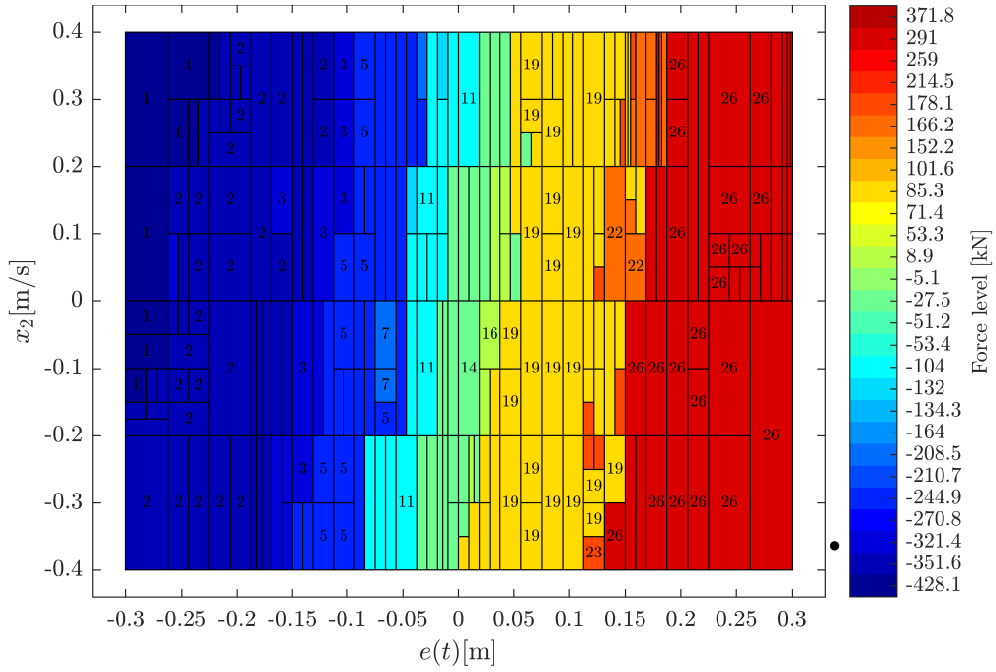


Figure 10.6. Two dimensional state space for eMPC position control. Initial force level is 3, corresponding to HML and marked with a bullet next to the colorbar. Colors note the first force level in the horizon of the approximated solution for each of the 264 hyper-rectangles.

In appendix E.1 on page 153 the solution for all 27 planes is shown. The tendency for all 27 solution planes is similar as seen on the two figures 10.5 and 10.6. The tendency shown is that around zero

error, the force levels close to zero are preferred. For positive errors larger force levels are required, and vice versa for negative error. Generally the solution planes are splitted vertically, with a small slant dependant on velocity.

Both figures clearly shows that some force levels are favoured for different initial force levels. Following the theory about MPC the algorithm should favour the more cost efficient shifts. Figure 10.7 shows the switching losses presented in section 5.5 on page 28 from the initial force levels used in the examples on figure 10.5 and 10.6. Comparing the shifts chosen by the algorithm with the loses for shifting losses it is noticed that favoured shifts matches the less expensive shifts in a region of force levels. Evaluating all solution planes tells that the algorithm never enters force levels 13 and 15, which are HHH and LLL respectively.

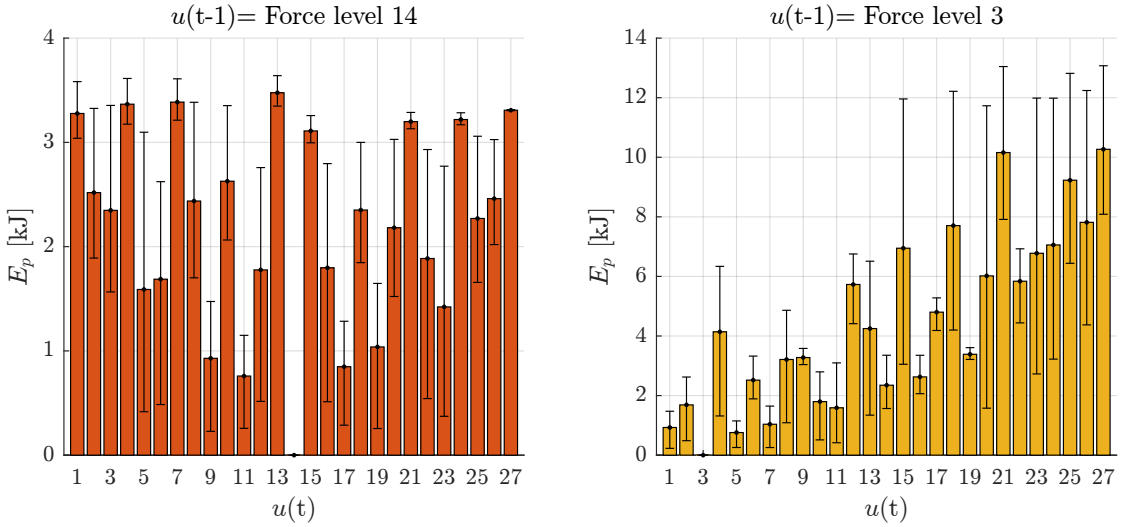


Figure 10.7. Losses from switching. From force level 14 to the left, and from force level 3 to the right. The bars shows the loss with a cylinder position of x_c = 1.1m. The error-bars shows the maximum and minimum losses for all cylinder positions in the range x_c=[0.2:1.8]m.

10.6.2 Two Dimensional State Space for Velocity Control

Here the solution for the reduced velocity problem is presented. Again 27 solution planes are found for each of the 27 force levels. Like for position control, the force levels 14 and 3 are used to illustrate the solution in figure 10.8 and 10.9 respectively. The number at each hyper-rectangle shows the first force level in the horizon of the approximated solution, and their colors corresponds to a force, shown on the colorbar to the right.

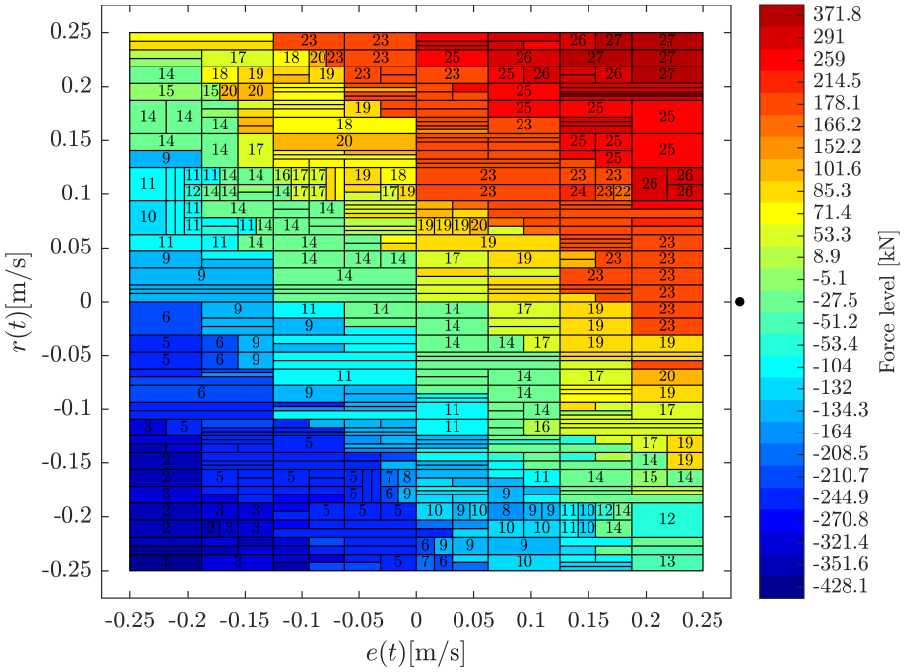


Figure 10.8. Two dimensional state space for eMPC velocity control. Initial force level is 14, corresponding to MMM and marked with a bullet next to the colorbar. Colors note the first force level in the horizon of the approximated solution for each of the 515 hyper-rectangles.

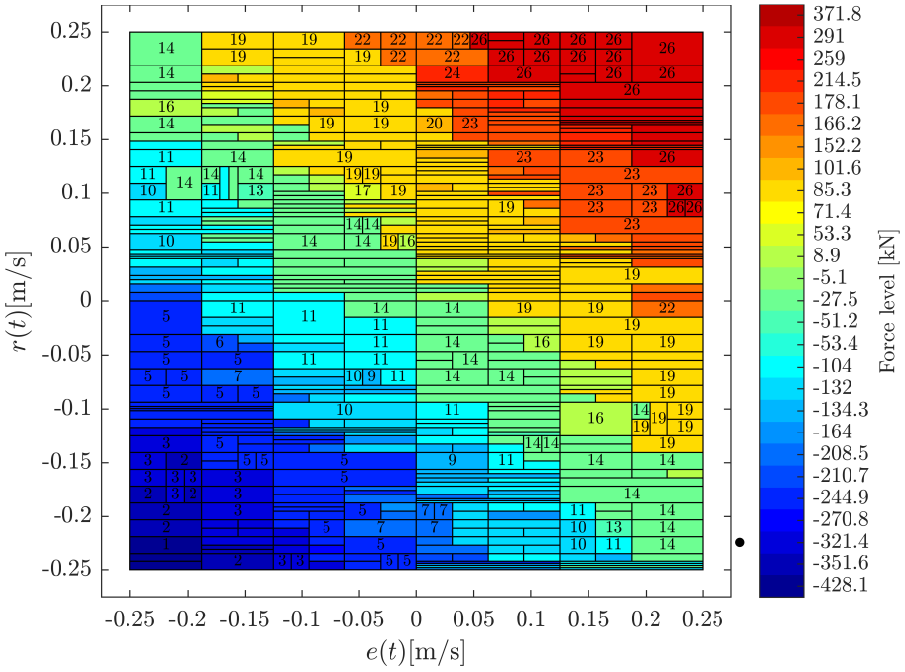


Figure 10.9. Two dimensional state space for eMPC velocity control. Initial force level is 3, corresponding to HML and marked with a bullet next to the colorbar. Colors note the first force level in the horizon of the approximated solution for each of the 548 hyper-rectangles.

All 27 solution planes are shown in appendix E.2. Common for all is a clear tendency along the diagonal from positive reference and negative error to negative reference to positive error. It is worth

noticing that in the diagonal, the close to zero force levels are preferred, especially 14 where all chambers are connected to medium pressure. Hence, if there is a large positive error and a large negative reference, the velocity is large and friction forces are large. Thus, the control utilises the friction to slow the velocity rather than forcing it.

Like for position control, some force levels a favoured. Again these matches the cheaper ones, with regard to switching loss as seen in figure 10.7.

10.7 Results

The reduced eMPC controller is evaluated on the full order model to investigate its potential as an alternative to the NMPC. The eMPC position controller is evaluated for a step response and its load holding capabilities. The eMPC velocity controller is evaluated for a step response and a sine wave reference. For each controller a RMS-error of the difference between reference and actual position or velocity is calculated, together with a total switching loss for the response period, calculated as in the input cost function in section 9.2.

10.7.1 Position Control

In figure 10.10 the eMPC position controller is tested for a step from 0.8m to 1.1m, with zero initial velocity. From the figure it is seen that the position response settles after approximately 2 seconds, with a small offset to the reference. The controller does only use a few force shifts to bring the position to the reference and keep it there. The eMPC is based on a constant control output for a small hyper-rectangle, which could explain the small offset from the reference. Thus, it is possible that a better tracking of the reference could be obtained by decreasing the error tolerances. The RMS-error and switching loss for the step response are e_{RMS}=150mm and E_{β}=3.825kJ respectively.

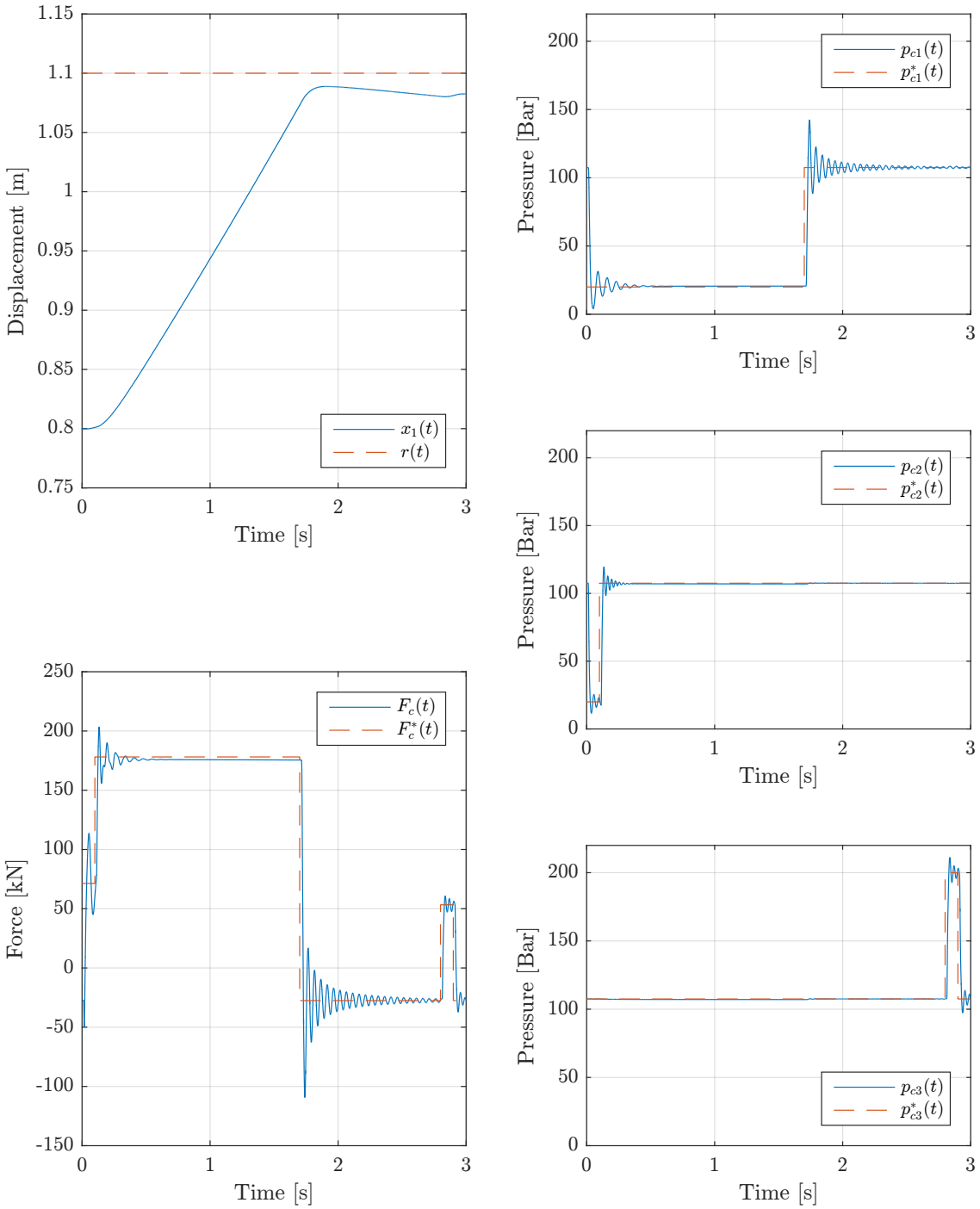


Figure 10.10. Step response with eMPC position control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

The load holding performance for the eMPC position controller is seen in figure 10.11. The initial velocity is zero and the initial position is 1.09m, which is slightly under the reference in order to simulate preceding history similar to the step response. An initial descrease in position is seen until a limit-cycle behaviour is reached. Worth noting is that only the reference to chamber three is changed, which is the smallest chamber. Hence, it is the most energy efficient shifts that are made. The RMS-error and the switching loss for the load holding response are e_{RMS}=17.1mm and E_β=7.499kJ respectively.

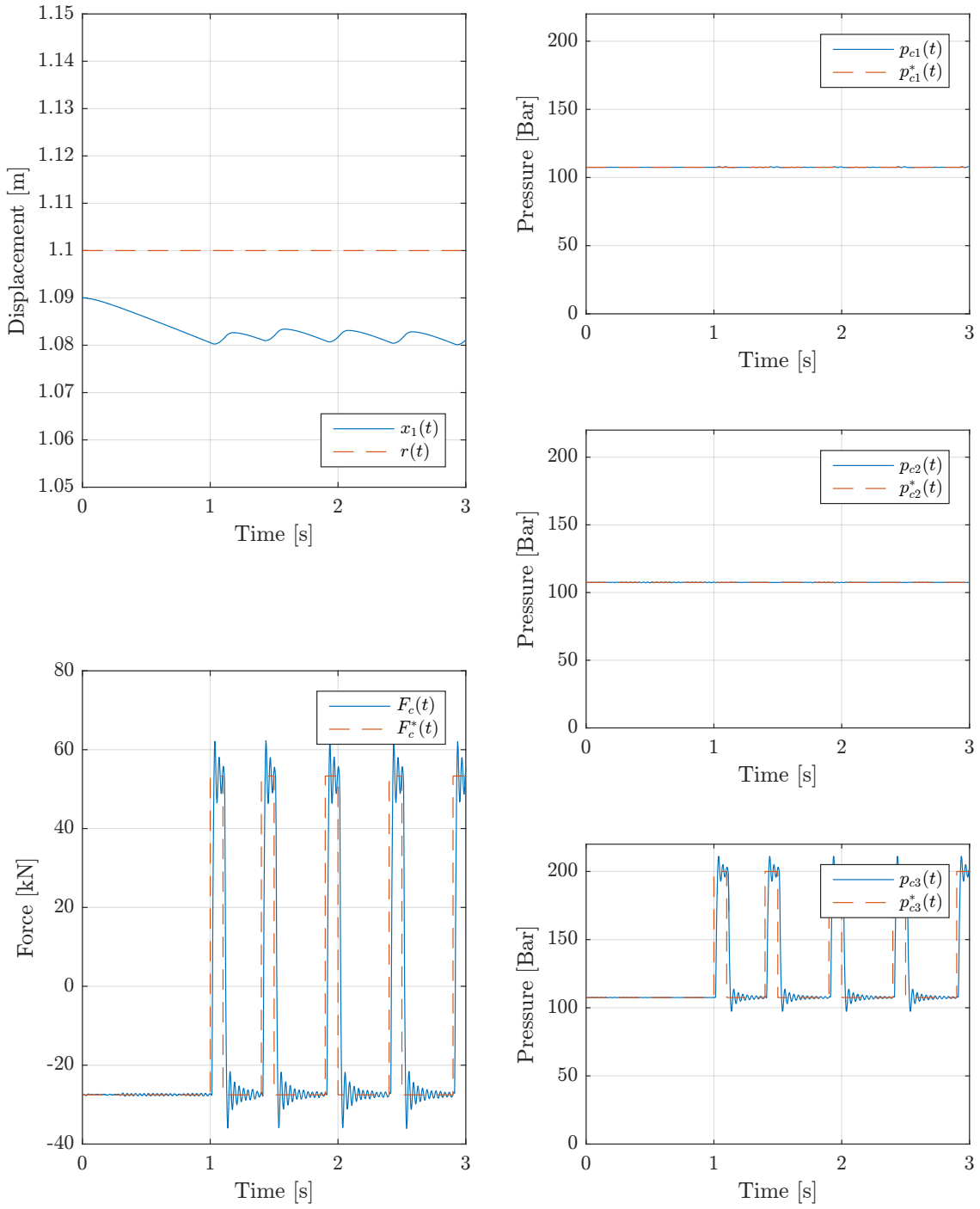


Figure 10.11. Load holding with eMPC position control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

10.7.2 Velocity Control

A velocity step response with the eMPC velocity controller is seen in figure 10.12. The step response has a slight overshoot due to large pressure oscillations in cylinder chamber 1. The velocity settles with a steady state error of approximately 0.4 m/s. This could be due to a large approximation error tolerance, when forming the explicit control law or due to the approximation of constant position when solving the eMPC velocity problem. Only two force shifts are used during this step response. The RMS-error and the switching loss for the step respone are e_{RMS}=63.2mm/s and E_β=2.348kJ

respectively.

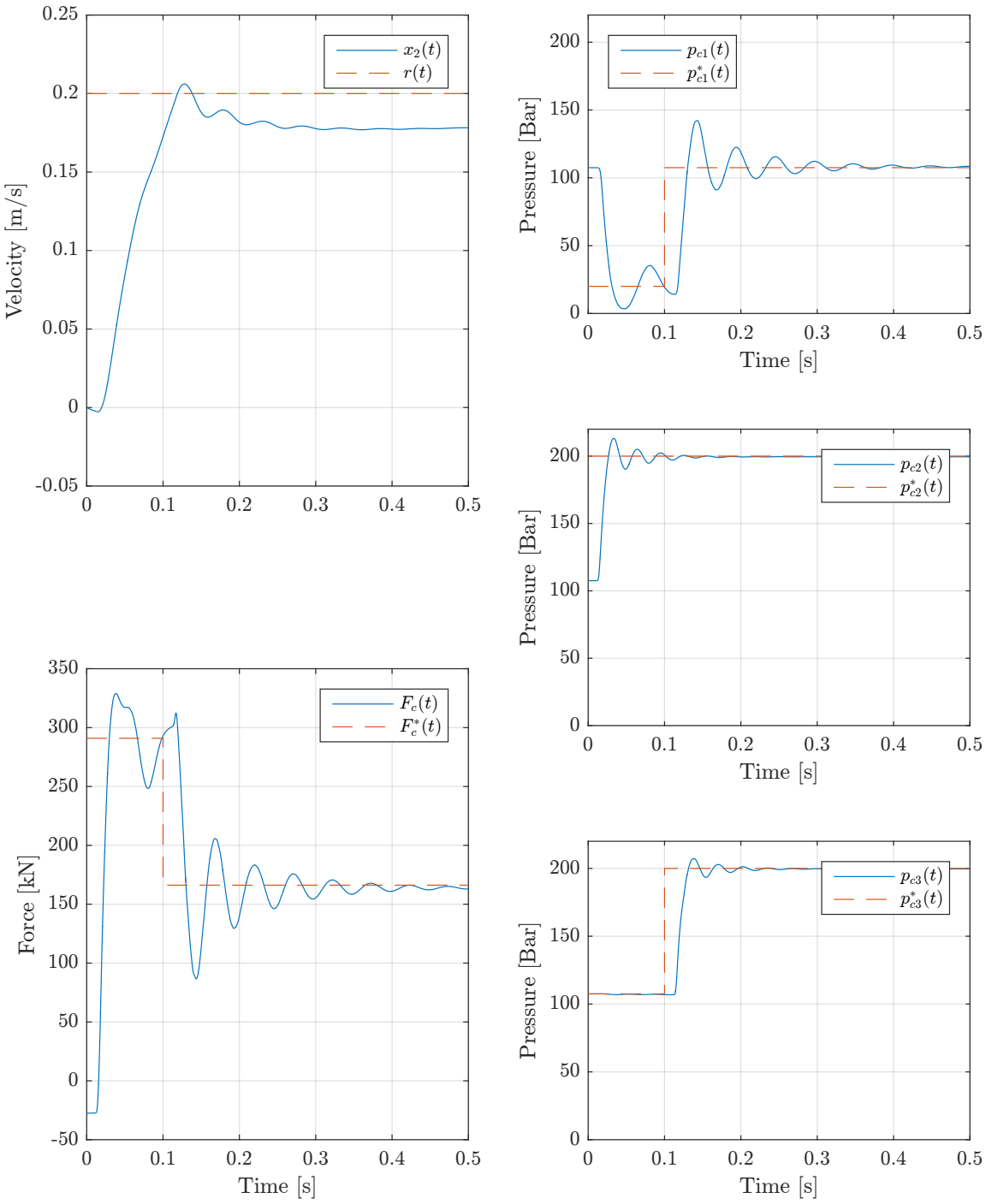


Figure 10.12. Step response with eMPC velocity control. Top left corner is velocity, bottom left is cylinder force and the three plots to the right are chamber pressures.

In figure 10.13 the eMPC is set to follow a sine trajectory. It can be seen that the velocity follows the reference with very little phase lag. It is seen that the smallest chamber 3 is most frequently switched, followed by the slightly larger chamber 2 with a slightly smaller amount of switching.

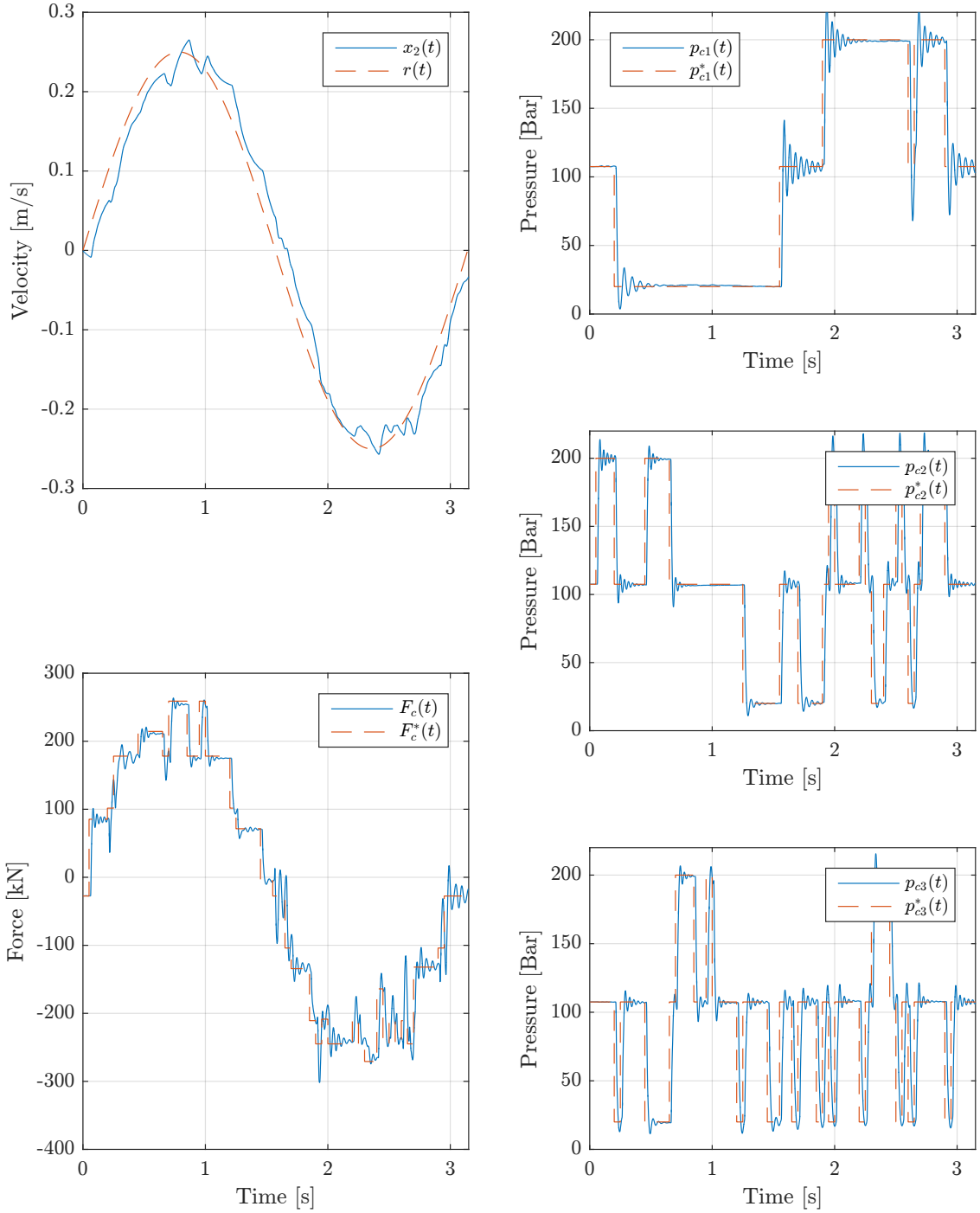


Figure 10.13. Sine wave reference trajectory with eMPC velocity control. Top left corner is velocity, bottom left is cylinder force and the three plots to the right are chamber pressures.

10.8 Chapter Summary

This chapter described the algorithm of converting an NMPC problem to an explicit form. A local constant approximation to the close-to-global solution is created for a series of hyper-rectangles covering the state space. In order to achieve a result before project deadline, the problem was reduced in dimension and size, and several modifications were done to increase the speed of convergence, here amongst reusing local optimal solutions point for later iterations.

Solutions were obtained for both position and velocity control. The solution to the position control problem was reduced to handle a constant reference. Simulated results shows a small steady state error, but with very low switching losses. The velocity control problem was reduced to a constant position, hence disregarding the position dependency of the switching losses. Similar to the position response, the velocity step response show a slight steady state error, again with very low switching losses.
The eMPC velocity control performed very well when subjected to a sine reference, with only a slight phase lag and few pressure level switching.

Both problem solutions was achieved with large approximation tolerances and a greatly reduced state space, and it is expected that performance can be increased with smaller tolerances. It is concluded that it is possible to convert an NMPC problem with a cost on switching losses to an explicit form in which the solution also penalises switching costs while minimising tracking error.

CHAPTER 11

LINEAR CONTROL

This chapter concerns the development of linear position and velocity control for the application case as well as the introduction of some control modifications to increase performance. A SISO approach is taken to simplify the development and the logic initially chosen is the simple minimum error logic, also known as linear quantization. Hence, the control output should be a force reference. Initially, a naive approach is taken, where the quantisation is linearised to a unity gain and thereby neglected in the linear control design. Considering the quantisation as a unity gain corresponds to the analysis in section 7.1.4, where it was shown that the quantizer tends to unity gain as the amount of quantization levels approaches infinity.

The mechanical system described by equation 7.20 is linearised by neglecting the Coulomb friction term and the gravitational term. The gravity term will not change significantly for small changes in position. It is considered reasonable to treat this as a static term, making it irrelevant for the dynamic analysis. As seen in figure 7.9 the equivalent mass range from approximately $70 \cdot 10^3$ to $180 \cdot 10^3$ kg, and hence an operating point must be chosen. It is chosen such to yield the smallest system eigenfrequency, which is when the equivalent mass is largest. To compare with eMPC, the PI controller is designed to work in the same range of positions. Hence, the largest mass is at $x_c = 1.4$, yielding a linearised mass of $m_{eq} = M_{eq}(1.4) = 100 \cdot 10^3$ kg. Figure 11.1 shows a block diagram of the linear system. The subscript $j = \{1,2\}$ are used to denote the controller and the mechanical system for position and velocity control respectively. The output y is either $y = x_c$ or $y = \dot{x}_c$ for position or velocity control respectively. The force reference is denoted u^* and the actual force applied to the mechanical system is denoted u.

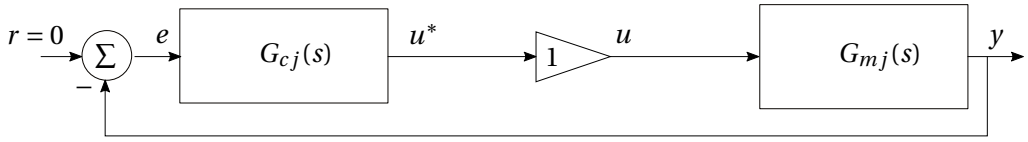


Figure 11.1. Block diagram of the linear system.

11.1 Position Control

The input-output relation for position control can be expressed as

$$G_{m1}(s) = \frac{x_c}{u} = \frac{1}{m_{eq}s^2 + B_{vs}s} \tag{11.1}$$

For position control a PI controller is implemented with the form

$$G_{c1}(s) = \frac{u^*}{e} = K_{c1}\frac{s+\omega_{c1}}{s} \tag{11.2}$$

The closed loop system then becomes a third order system with a zero expressed as

$$G_{s1}(s) = \frac{G_{c1}(s)G_{m1}(s)}{1+G_{c1}(s)G_{m1}(s)} = \frac{K_{c1}(s+\omega_{c1})}{s^3 + \frac{B_{vs}}{m_{eq}}s^2 + K_{c1}s + K_{c1}\omega_{c1}} \tag{11.3}$$

The control parameters are determined by evaluating the step response of the system. The control parameters for the position controller are tuned to obtain similar step response as the response obtained with the NMPC controller seen in section 9.4. The controller is tuned to obtain similar step response for better comparison between the control performance of the NMPC, eMPC and the linear controller. Furthermore, the control gain is designed such that output does not reach saturation as long as the position error is less than 0.5 m. The step response of the system and the corresponding bode plot for the open loop uncompensated and compensated system are shown in figure 11.2 and 11.3 respectively.

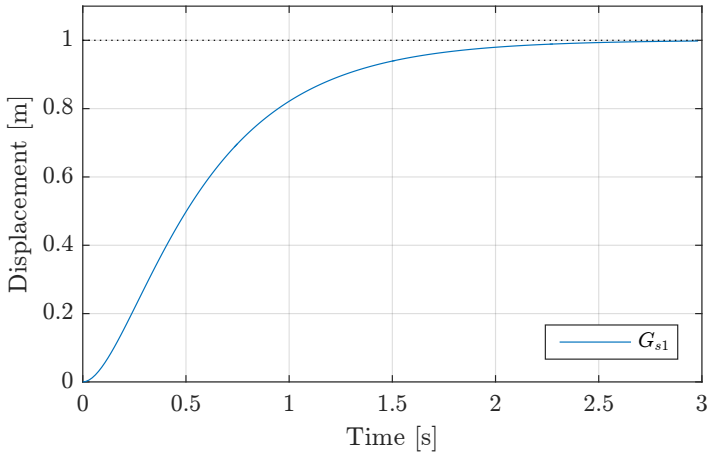


Figure 11.2. Positionon step response of the closed loop system with PI controller.

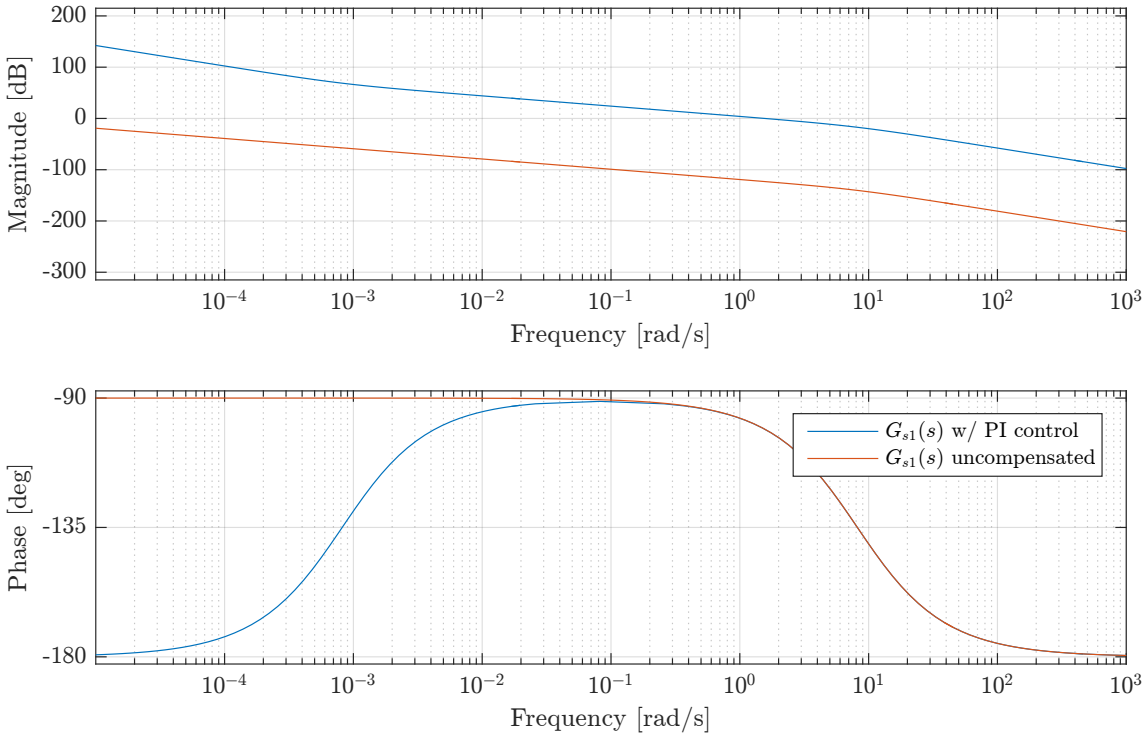


Figure 11.3. Bode plot of the open loop uncompensated system and open loop system with PI controller.

The zero of the PI controller is placed four decades before the mechanical pole. The system has a settling time of approximately 2 seconds and no overshoot. The open loop system has a phase margin of 83.3° and an infinity gain margin. Hence, the linear system is at the operating point stable.

11.2 Velocity Control

The input-output relation for velocity controlled system is

$$G_{m2}(s) = \frac{\dot{x}_c}{u} = \frac{1}{m_{eq}s + B_{vs}} \tag{11.4}$$

For velocity control a PI controller is implemented with the form

$$G_{c2}(s) = \frac{u^*}{e} = K_{c2}\frac{s+\omega_{c2}}{s} \tag{11.5}$$

The closed loop system is then expressed as

$$G_{s2}(s) = \frac{G_{c2}(s)G_{m2}(s)}{1+G_{c2}(s)G_{m2}(s)} = \frac{K_{c2}(s+\omega_{c2})}{s^2 + \left(\frac{B_{vs}+K_{c2}}{m_{eq}}\right)s + \frac{K_{c2}\omega_{c2}}{m_{eq}}} \tag{11.6}$$

The closed loop system is a second order system with a zero. The eigenfrequency and damping of the system can be expressed as

$$\omega_n^2 = \frac{K_{c2}\omega_{c2}}{m_{eq}} \tag{11.7}$$

$$2\zeta\omega_n = \frac{B_{vs}+K_{c2}}{m_{eq}} \tag{11.8}$$

As for the position controller the control parameters are tuned to obtain a similar step response as the NMPC seen in section 9.4. The velocity step response is seen in figure 11.4, while the corresponding bode plot of the open loop uncompensated and compensated system is shown in figure 11.5.

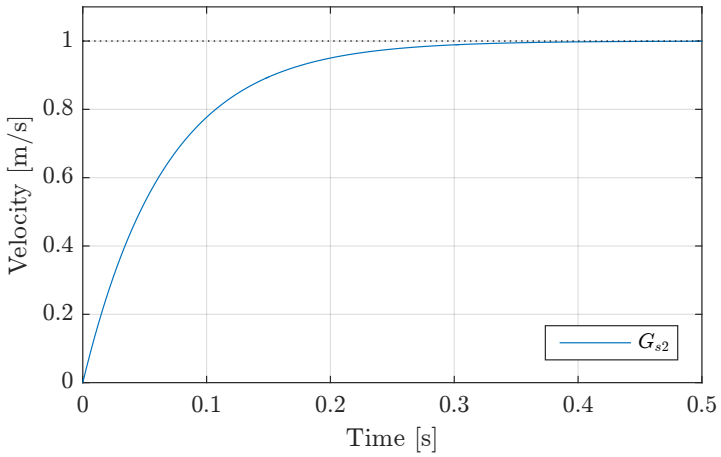


Figure 11.4. Velocity step response of the closed loop system with PI controller.

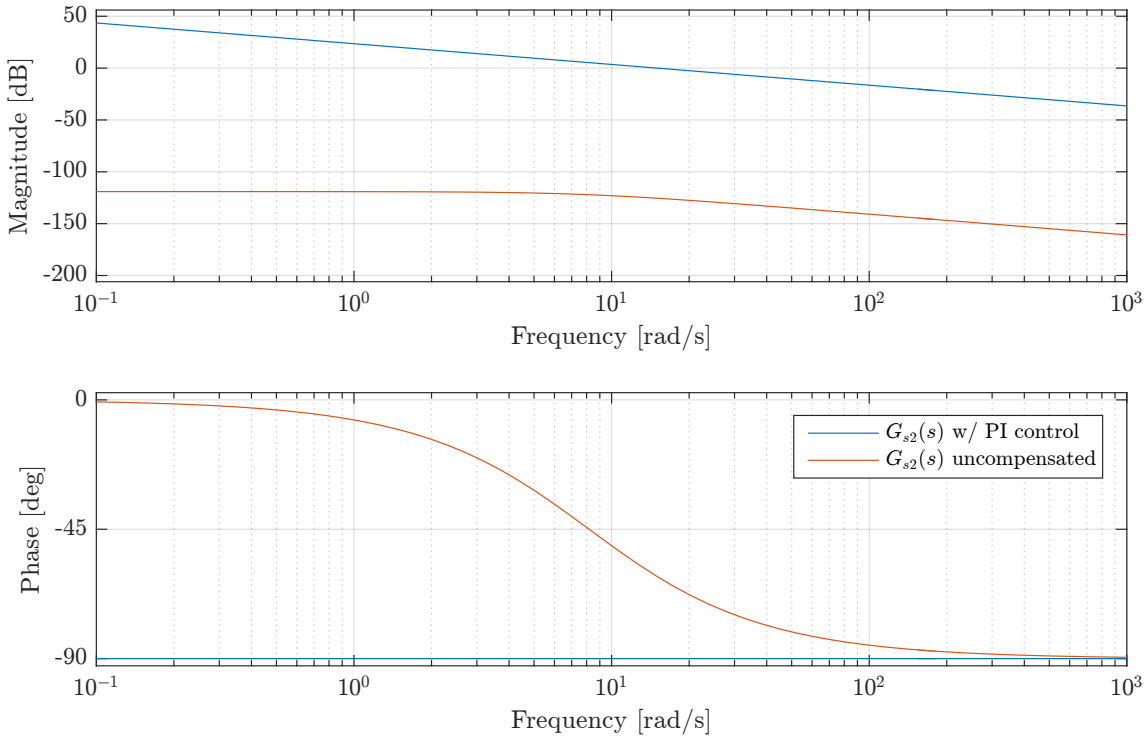


Figure 11.5. Bode plot of the open loop uncompensated system and open loop system with PI control.

The zero of the PI controller is placed at the mechanical pole, when it is at its minimum, which is when $m_{eq} = 100 \cdot 10^3$ kg. A unity damping factor and a settling time of approximately 0.3 second are achieved. The open loop phase and gain margin are 90° and infinity respectively. Hence, the linear system is stable at the operating point.

11.3 Anti Wind-up

As was shown in section 7.1.4, quantisation can induce oscillations or limit cycles in the system. One way to address this problem can be to introduce anti wind-up (AW) in a similar manner as for saturation non-linearities. A block diagram of the control system including AW is shown in figure 11.6.

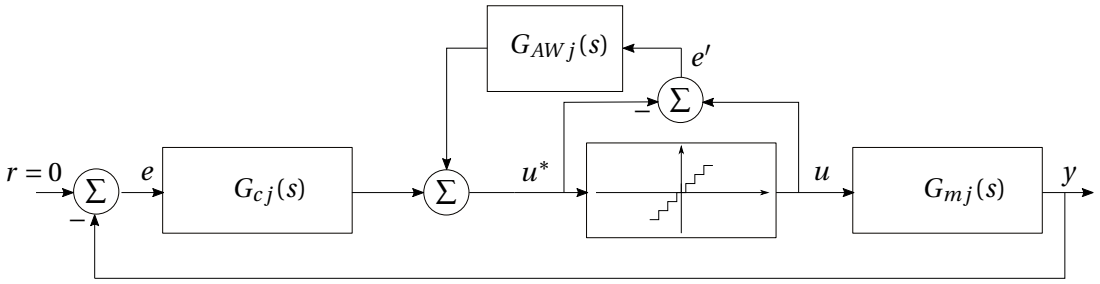


Figure 11.6. Implementation of AW in a system with quantised inputs.

11.3.1 Anti Wind-up for Velocity Control

In the following, AW is designed for the velocity controller. The control output is a function of e and e':

$$u^* = G_{c2}(s)e + G_{AW2}(s)e' \tag{11.9}$$

where

$$G_{c2}(s) = K_{c2}\frac{s+\omega_c}{s} \tag{11.10}$$

$$G_{AW2}(s) = K_{AW2}\frac{1}{s} \tag{11.11}$$

where e' denotes the quantisation error. e' can be interpreted as an output disturbance and it is therefore relevant to analyse this disturbance, and how the system reacts to it. The following relations can be deduced from figure 11.6

$$e = -\dot{x}_s \tag{11.12}$$

$$e' = u - u^* \tag{11.13}$$

$$x = G_{m2}(s)u \tag{11.14}$$

From the above a relation between u^* and e' can be derived, yielding

$$G_{eu2}(s) = \frac{u^*}{e'} = \frac{G_{AW2}(s) - G_{c2}(s)G_{m2}(s)}{1 + G_{c2}(s)G_{m2}(s)} \tag{11.15}$$

Which has the steady state output of

$$G_{eu2}(0) = \frac{K_{AW2}B_{vs}}{K_{c2}\omega_{c2}} - 1 \tag{11.16}$$

Thus, for $K_{AW2} < \frac{K_{c2}\omega_{c2}}{B_{vs}}$, $G_{eu}(0) < 0$ and a positive disturbance e' will result in u^* becoming negative. This response to disturbances is natural for most disturbances. However, in the case of a quantizer, this will result in a jump down in u, which then results in a negative e' leading to a positive u^*. Instead if $K_{AW2} = \frac{K_{c2}\omega_{c2}}{B_{vs}}$, $G_{eu}(0) = 0$ and the control 'ignores' the quantisation disturbance. Since this

disturbance is predictable, this will not effect general disturbance rejection, but will result in a steady state error, that is maximum when the quantisation error is maximum, which is half the quantisation increment size, Δ_i, $e'_{max} = 0.5\Delta_i$. The maximum steady state error caused by quantisation is then

$$e_{m,max} = G_{m2}(0)e'_{max} = \frac{\Delta_i}{B_{vs}} \tag{11.17}$$

Subscript i here reflects a non-uniform step size of the quantisation. Since B_{vs} is not known exactly, K_{AW} should be designed with some margin. For the case system, the force range of the DDC is close to 800 kN. Assuming evenly spaced step size of the quantisation, the average step is 800/27 kN, which yields a maximum error of 14.8 kN. With a viscous friction of $90 \cdot 10^4$ N/(m/s), the steady state error in velocity is at maximum 0.016 m/s. Hence, for this system, the maximum steady state velocity error is considered acceptable.

11.3.2 Anti Wind-up for Position Control

For position control, the system G_{m1} contains a free integrator and with a similar analysis as for the velocity control AW it is found that

$$G_{eu1}(s) = \frac{u^*}{e'} = \frac{G_{AW1}(s) - G_{c1}(s)G_{m1}(s)}{1 + G_{c1}(s)G_{m1}(s)} \tag{11.18}$$

where the steady state gain is

$$G_{eu1}(0) = -1 \tag{11.19}$$

Hence, a limit cycle behaviour can not be avoided with this type of AW and this problem must be addressed with other tools. As a final remark on AW for position control, it was found that by varying K_{AW1} it was possible to alter the frequency of the limit cycle. However, further analysis of equation 11.18 yielded no conclusive result.

11.4 Results

The linear controllers are evaluated by simulating the response on the full order model. First, the linear position controller is evaluated for a step response and for its load holding capabilities. Second, the linear velocity controller is evaluated for a step response, both with and without anti wind-up. The linear velocity control is also tested for a sine reference.
Both controllers are used with a linear quantisation coupled with a minimum switching period, t_min, for switching between pressure lines. For position control t_min = 0.1s, while for velocity control t_min = 0.05s. This is chosen for better comparability to the MPC controllers.

Similar to the other controllers an RMS-error of the tracking is calculated, together with a total switching loss for the response period, calculated as in the input cost function in section 9.2.

11.4.1 Position Control

The linear position control is tested for a step from 0.8m to 1.1m in figure 11.7. It can be seen that the control uses many force steps before settling at a close to minimum force output. This results in a high number of switching between line pressure for each chamber. The position error is close to zero after 1.5 seconds, but thereafter converges slowly. The RMS-error is, e_{RMS}=110.8mm and the switching loss totals to E_β=35.86kJ.

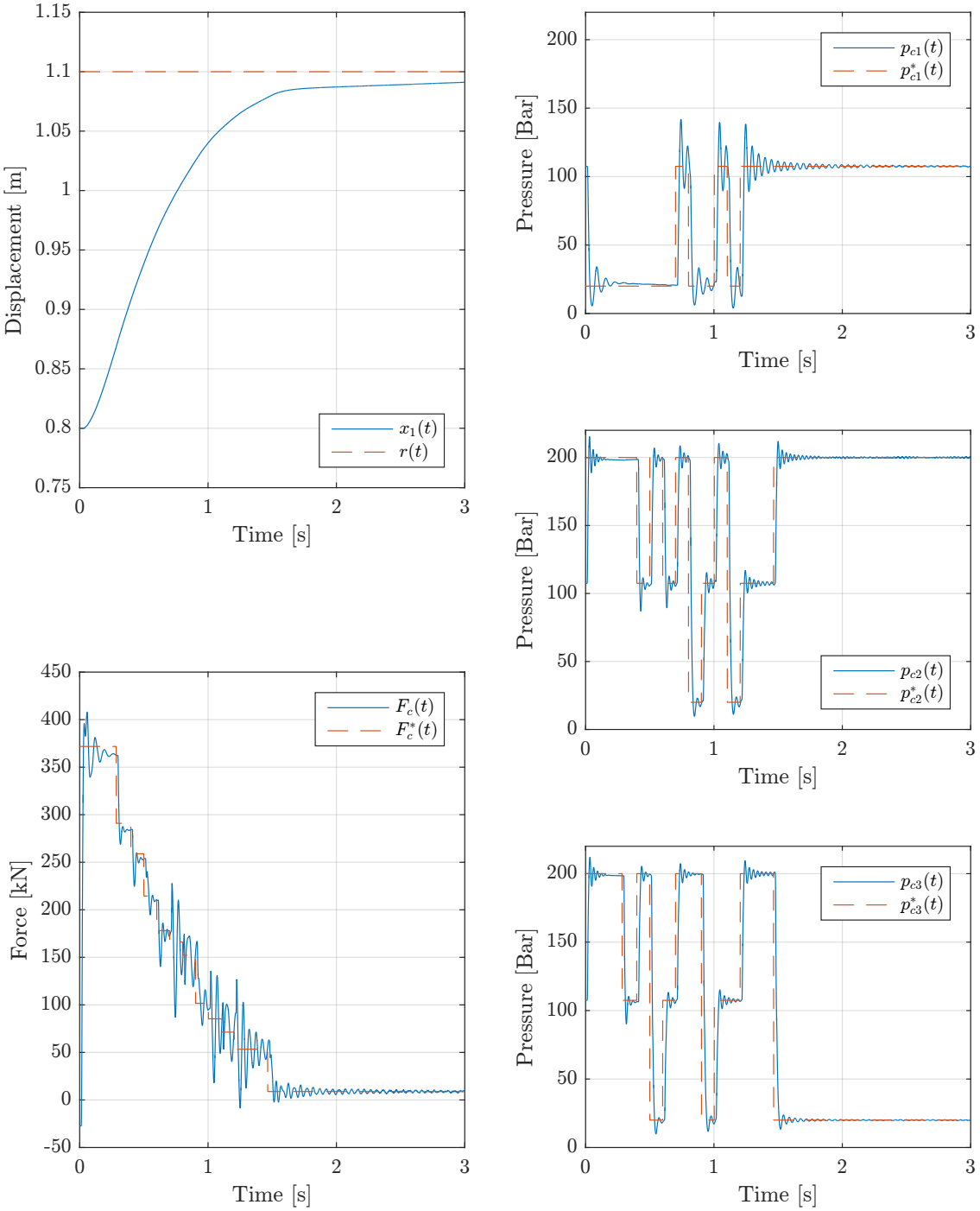


Figure 11.7. Step response with linear PI position control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

The load holding performance for the linear position control is seen in figure 11.8. A slight error of 0.01m is introduced to simulate a preceding history similar to the step response. It can be seen, that in this point the linear position control handles load holding very well and no limit cycles are seen. This could be due to a fortunate combination between static forces and quantisation error yielding a very little acceleration. The RMS-error is, e_{RMS}=5.4mm and the switching loss is E_β=5.35kJ.

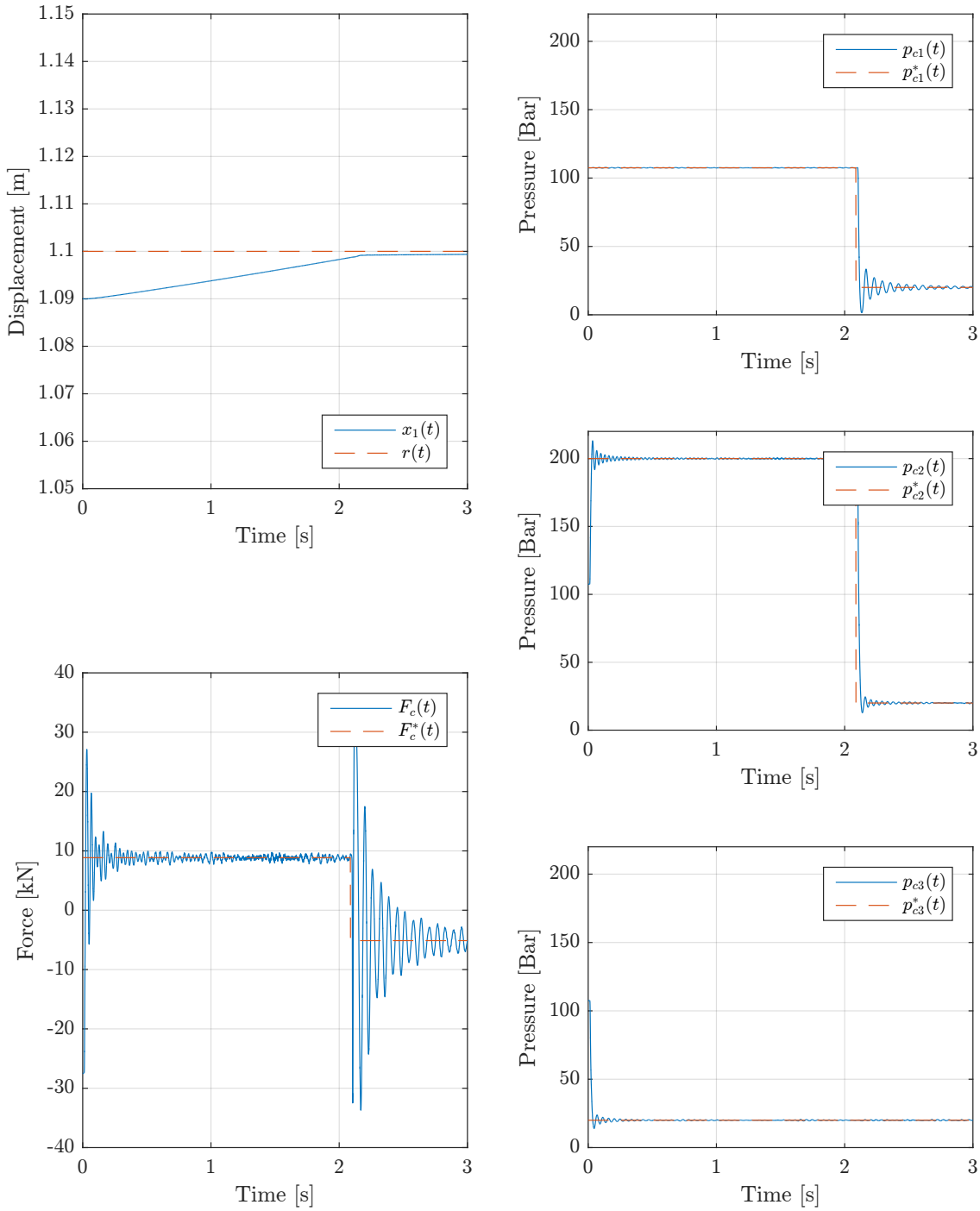


Figure 11.8. Load holding with linear PI position control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

11.4.2 Velocity Control

Initially, the linear velocity control is tested with no anti wind-up in figure 11.9. It is tested for a step response from 0 to 0.2m/s at center position. It can be seen that the velocity error is zero after 0.15 seconds, but then enters a limit-cycle with frequent switching. Resultantly, the RMS-error is e_{RMS}=61.8mm/s and the switching loss is E_β=22.73kJ.

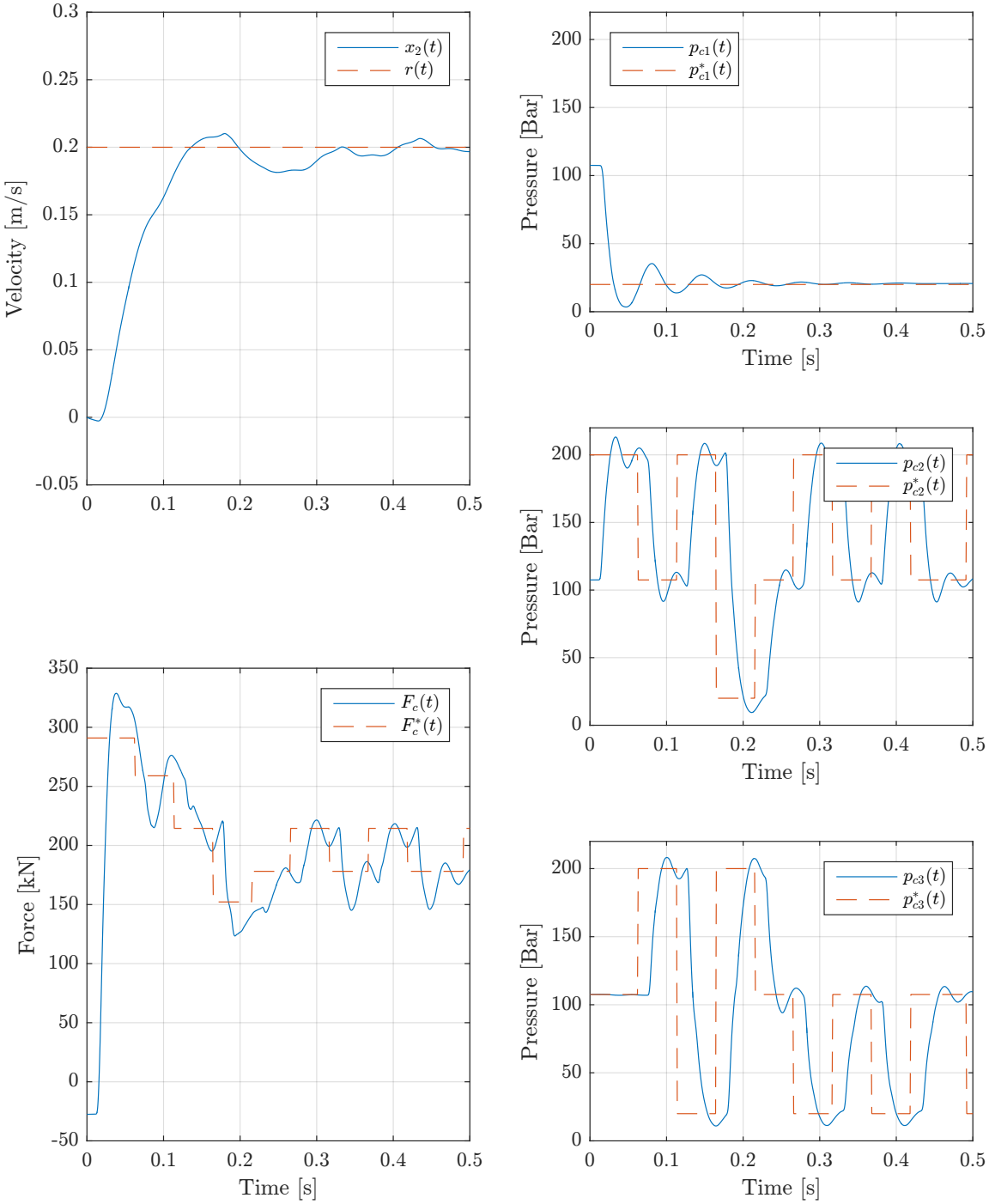


Figure 11.9. Step response with linear PI- Velocity control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

Figure 11.10 show the step response for the linear velocity control with added anti wind-up. The initial behaviour is similar for the two controllers, but the linear control with AW does not enter a limit-cycle. Instead, it settles with a small steady state error as intended. The RMS-error is e_{RMS}=62.0mm/s and the switching loss is E_β=7.35kJ. It is concluded from here, that the anti wind-up functions as intended and is included for the remainder of this thesis.

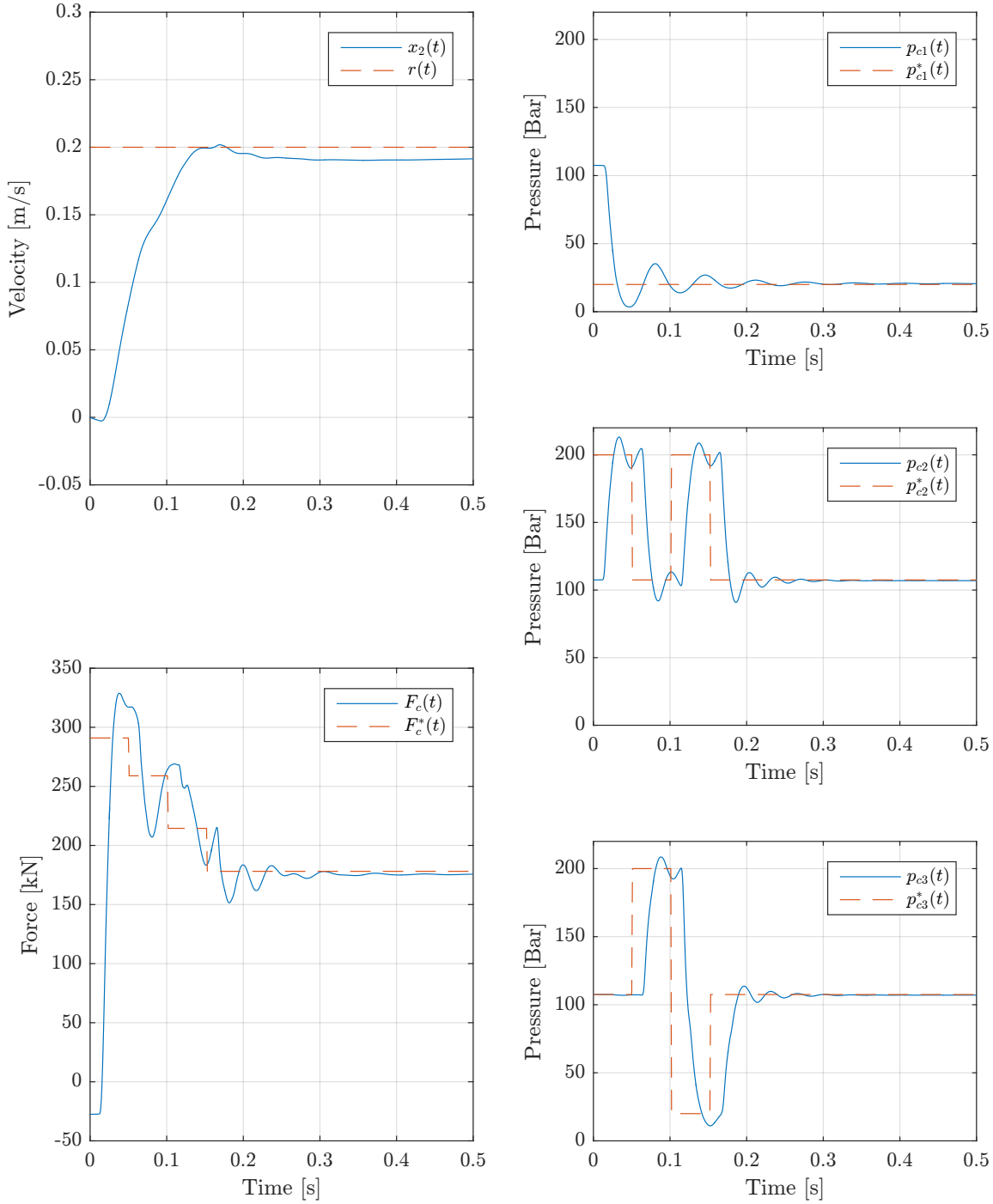


Figure 11.10. Step response with linear PI velocity control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

In figure 11.11 the linear velocity control with AW is set to follow a sine trajectory. The force shifts are frequent when the gradient of the trajectory is large, as the linear controller uses each force closest to the desired force. When the gradient is small, i.e. near the peak values, the force settles on a constant value. This is due to the anti wind-up, and the effect on the error is minimal. The RMS-error is e_{RMS}=33.2mm/s and the switching cost is E_β=155.684kJ.

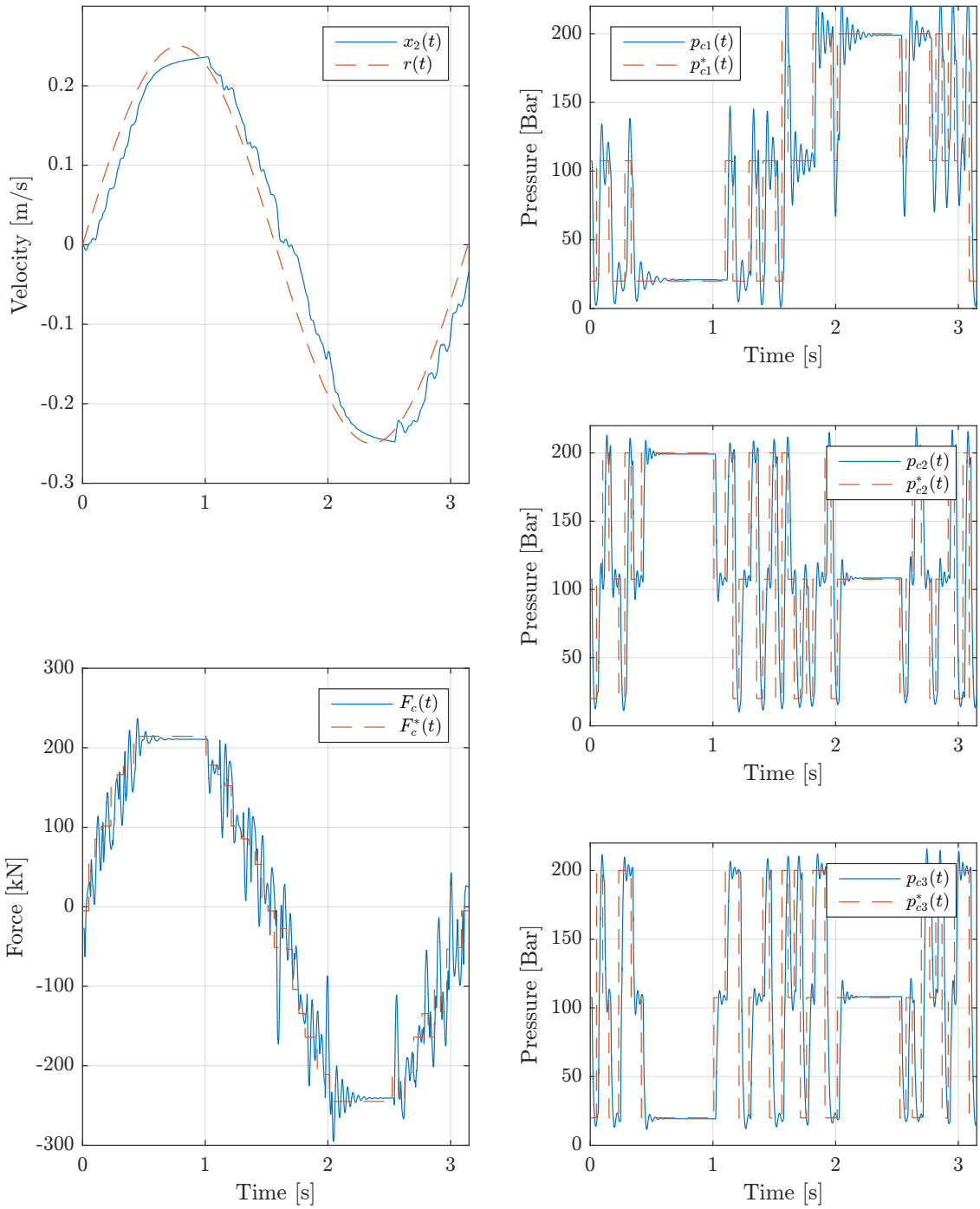


Figure 11.11. Sine wave reference trajectory with linear PI velocity control. Top left corner is position, bottom left is cylinder force and the three plots to the right are chamber pressures.

11.5 Chapter Summary

Linear position and velocity control have in this chapter been developed and tested on the full order model. Generally many force shifts are seen with the linear controllers, as they chooses the force reference closest to the desired force. For PI velocity control an AW has been developed and implemented. Simulated results with AW for a step response and a sine wave reference has shown good results compared to simulated results without AW. With the AW implemented a small steady

state error is obtained instead of a limit-cycle around the reference. Thus, the implemented AW function as intended.

Part III

Comparison & Final Remarks

CHAPTER 12

COMPARISON

The NMPC, eMPC and linear control results obtained in the previous chapters, will in this chapter be compared to each other. It is not possible to use the NMPC in a real-time implementation, due to too large computational requirements. However, the NMPC is used for comparison, since it can indicate the performance limits of the eMPC, based on the assumption that with increased computational power, the eMPC performance can be made close to the NMPC, by decreasing the error bounds and thereby increasing the resolution of the eMPC solution. The position controllers are compared with respect to the step response and the load holding capability, while the velocity controllers are compared with respect to the step response and the sine reference tracking. Furthermore, the RMS-error and the switching losses are compared for each response. Lastly the robustness to change in friction and line pressures are compared for the three controllers.

12.1 Position Control Comparison

In figure 12.1 the previous obtained step response and force reference for NMPC, eMPC and PI position control are shown. From the figure it is seen that the eMPC has the slowest response and a slightly larger offset compared to the PI controller.

The difference between the PI controller and the two other controllers is clearly seen in the force reference. As previous described the PI controller chooses the force reference closest to the desired force, which result in shifts to all force levels between the maximum and minimum desired force reference, whereas the MPC controllers chooses the force levels based on a balance between tracking error and switching loss. The capability of balancing tracking error and switching loss become evident when comparing the RMS-error and total switching losses, which are seen in figure 12.2. The lowest RMS-error is obtained with PI control, while eMPC has the largest RMS-error, which is 35.4% greater than the RMS-error with PI control. However, comparison of switching losses reveal that the PI controller has a significant larger switching loss compared to the two MPC controllers. Compared to the switching loss with PI control, the switching loss is reduced 72.2% and 89.4% for NMPC and eMPC respectively.

A comparison of the load holding capability for the three controllers and the corresponding force reference are shown in figure 12.3. The resulting RMS-errors and switching losses can be seen in 12.4. It can be seen that similar RMS-errors are achieved with NMPC and PI-control, while the steady state error of the eMPC results in a larger RMS-error.

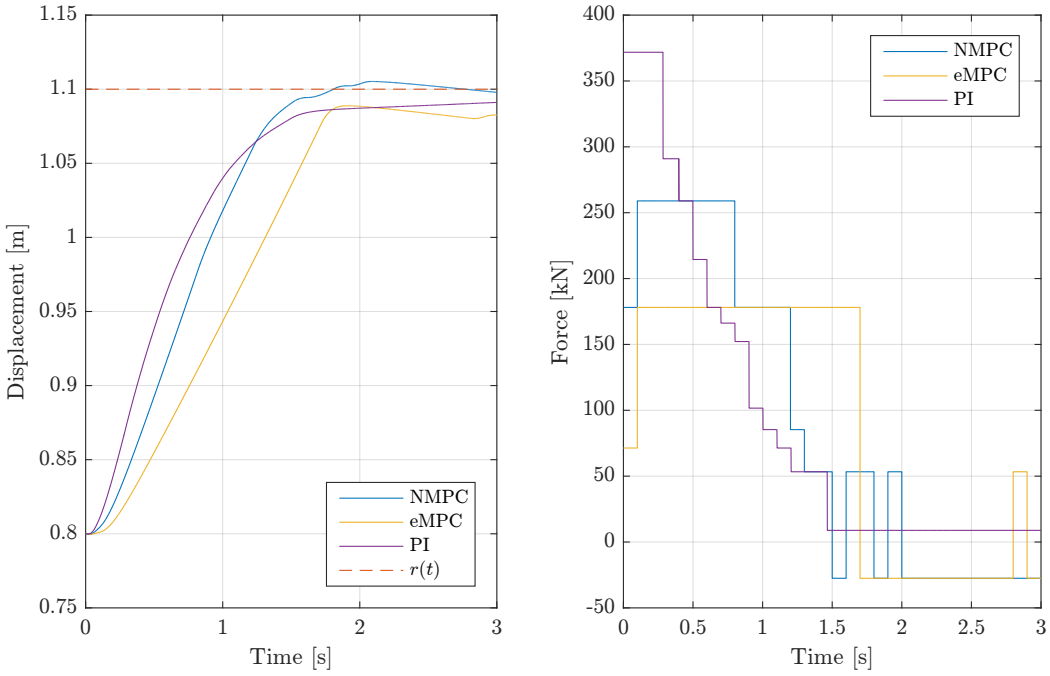


Figure 12.1. Position response and force reference for a position step with NMPC, eMPC and PI position control.

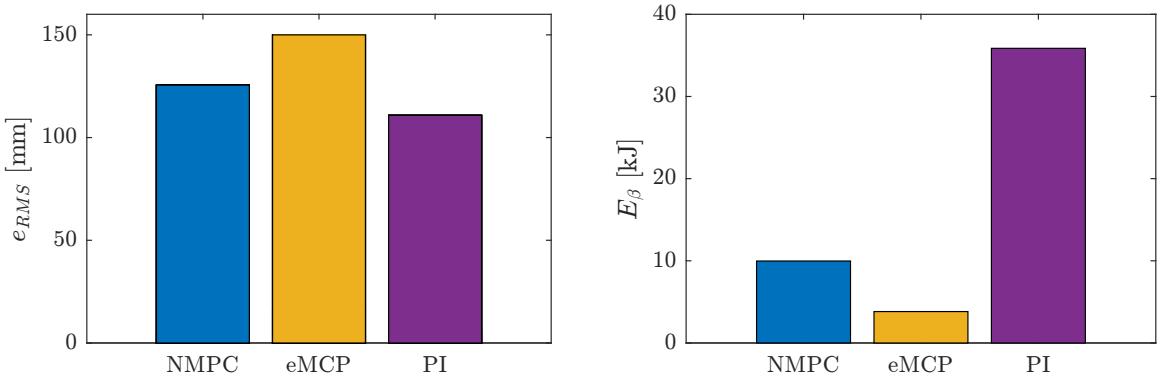


Figure 12.2. RMS-error and switching losses for NMPC, eMPC and PI position control for position step response.

Interestingly, the switching losses are similar for all three controllers, even though the PI control switches only two times. This illustrates that the number of switching is not necessarily a good estimator for switching losses.

The above comparison show that it is indeed possible to achieve a low position error, while considering the losses associated with switching. The NMPC and eMPC both perform well compared to the standard PI control. While eMPC control is not performing exactly similar to the NMPC from which it is derived, the switching losses are still greatly reduced while still maintaining a small control error. It is considered reasonable to expect that the performance differences between NMPC and eMPC can be decreased by decreasing tolerances of the approximation algorithm.

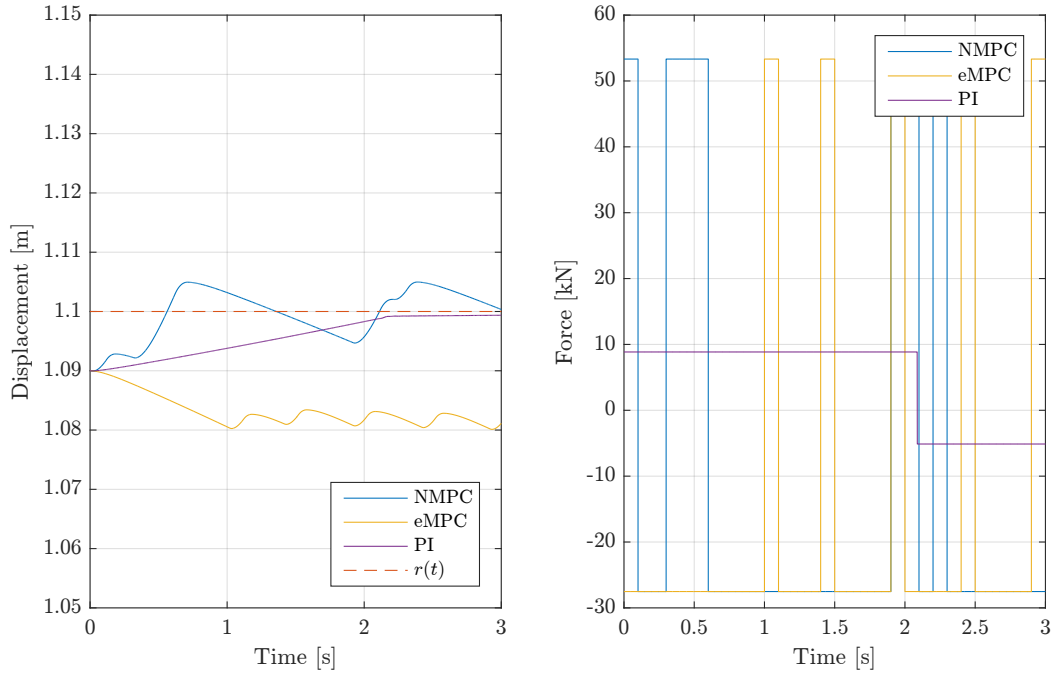


Figure 12.3. Position response and force reference for load holding with NMPC, eMPC and PI position control

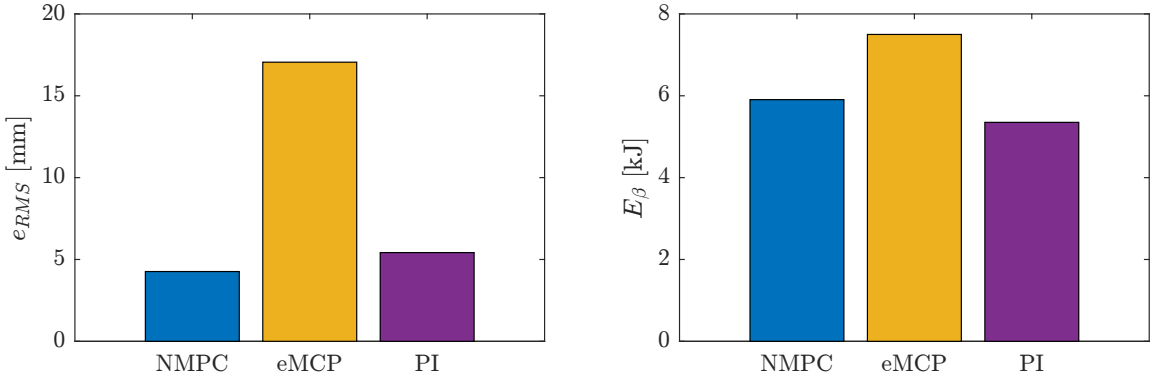


Figure 12.4. RMS-error and switching losses for NMPC, eMPC and PI position control for load holding.

12.2 Velocity Control Comparison

The velocity step response for the three controllers are seen in figures 12.5. Similar step responses are seen, while the PI and eMPC have a slight steady state error. The force output of the three controllers are also similar, but with PI control using more force levels.

The resulting RMS-errors and switching losses are shown in figure 12.6. It is seen that the RMS-error are nearly the same for all three controllers. However, the switching losses reveal much larger losses for the PI control. Compared to the switching loss with PI control, the switching losses with eMPC is reduced by 68%. Hence large energy reduction is achieved with similar tracking performance.

For the sine trajectory, the tracking performance of the three controllers are indistinguishable and all have similar phase lag. The control output are equally similar, but with a less varying control output from the PI-control.

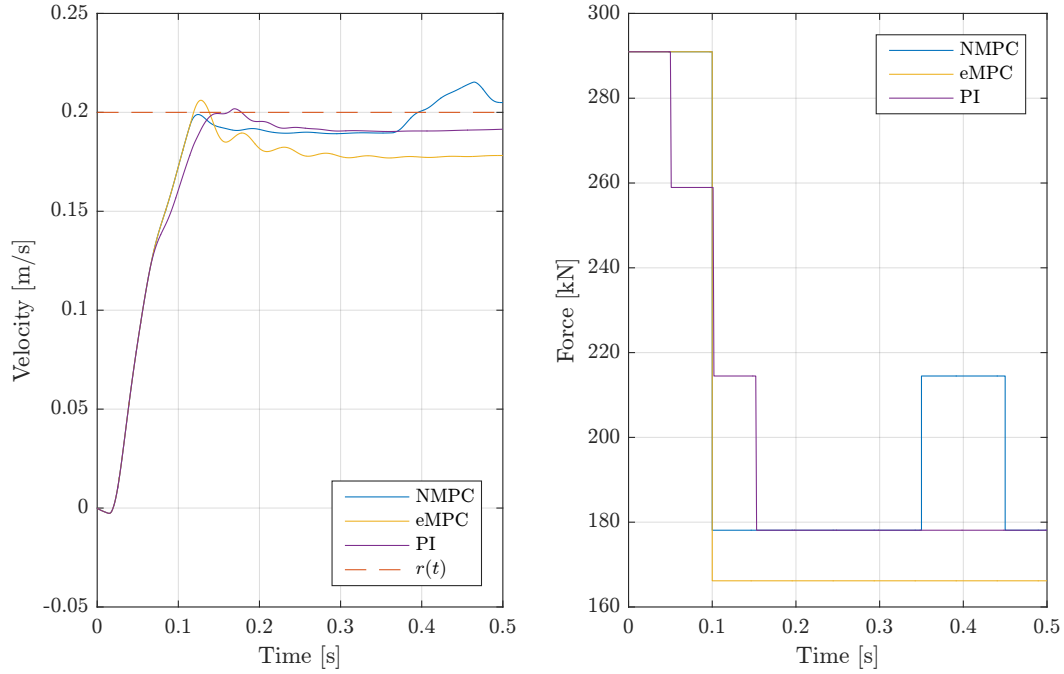


Figure 12.5. Velocity response and force reference for a velocity step with NMPC, eMPC and PI velocity control.

The RMS-errors are also similar for the three controllers, with the PI control having a larger error than the two MPC controllers. As was generally the case with position control, for velocity control the switching losses are considerably reduced using MPC controllers. Here eMPC both RMS-error and switching losses are reduced by 19.9% and 66.7% respectively compared to PI control.

have clearly the smallest losses, while the PI-control has the largest losses. The results show that like

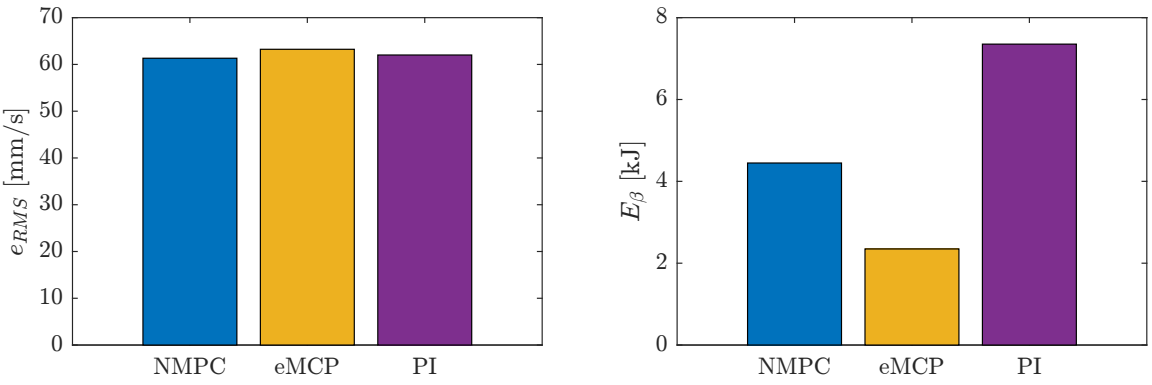


Figure 12.6. RMS-error and switching losses for velocity step with NMPC, eMPC and PI velocity contro.

for position control, an NMPC can be designed such to penalise switching losses while minimising velocity error. Furthermore, the explicit implementation yields good results while reducing switching losses. As was the case for MPC position control, NMPC and eMPC yield dissimilar results, but both manage to reduce switching losses and have tracking performance which is comparable or better than a PI control.

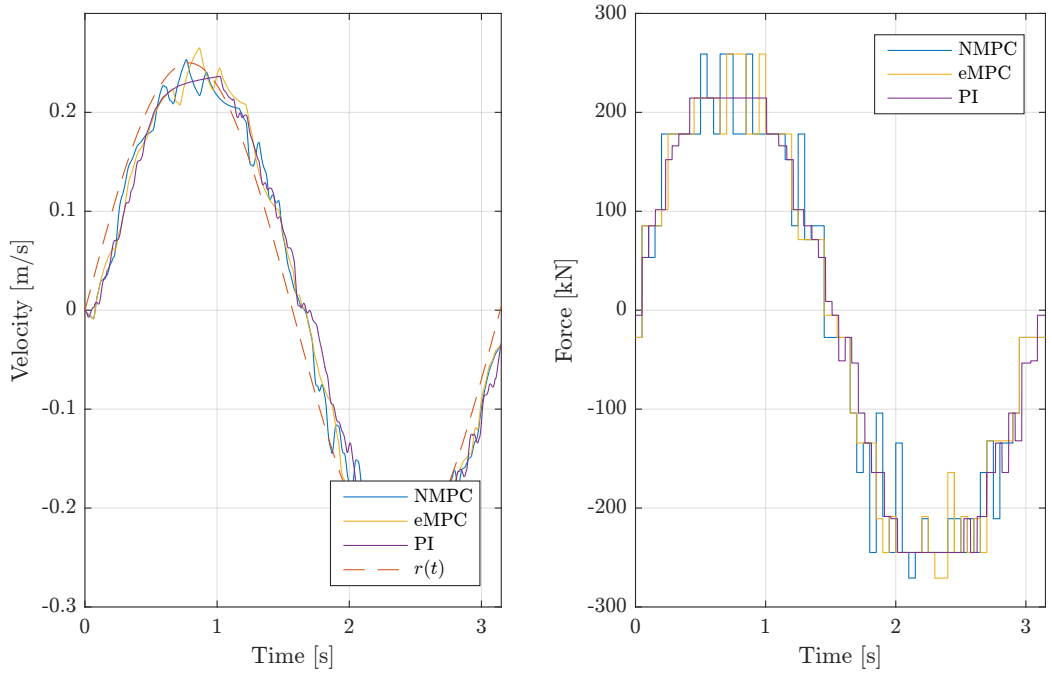


Figure 12.7. Velocity response and force reference for sine wave trajectory with NMPC, eMPC and PI velocity control.

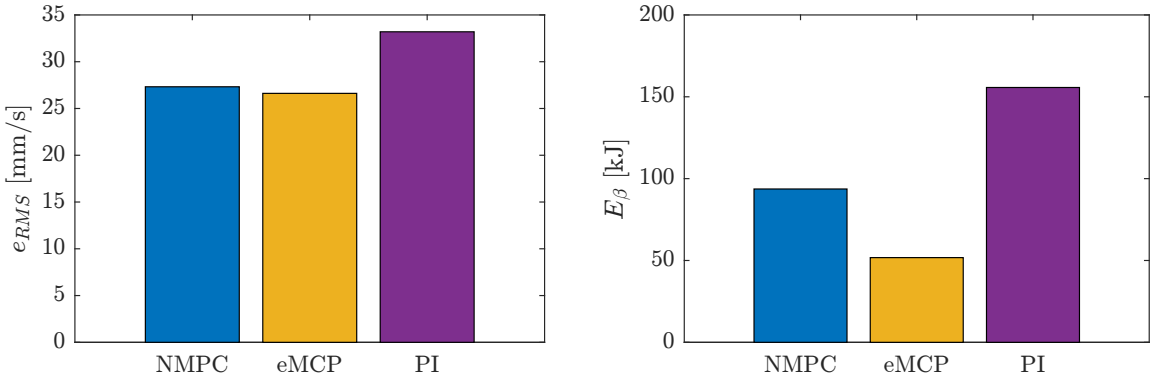


Figure 12.8. RMS-error and switching losses for sine wave trajectory with NMPC, eMPC and PI velocity control.

12.3 Robustness Analysis

The basis for MPC is a refined knowledge about the system. Thus, the robustness to change in system parameters is evaluated for NMPC and eMPC. This is done by using the same control parameters as found in section 9.4 on page 60, after which some of the system parameters are changed. Again the NMPC and eMPC are compared together with the linear PI controller, also using the same control parameters as found in chapter 11 on page 93.

First the robustness to change in friction is investigated. This is chosen because friction is a parameter that often is hard to determine and in most application friction can change over time as joints and bearings erode. To evaluate the robustness to friction, it is changed by ±20%. The next parameter investigated is the line pressure, p_S=[p_H p_M p_L]. Even though the line pressures could be input to the NMPC, as the NMPC indicate the performance limits of the eMPC. By changing the line pressure

the predicted force by the NMPC and eMPC no longer corresponds to the actual force. Robustness for changes in line pressure is tested for ±10%.
Generally the switching losses are not changed notable for the different changes in friction and line pressures. Hence, the force shifts and the associated switching losses will not be presented.

Figure 12.9 show the position step response for all three position controllers, where the friction parameters are changed in the full order model. The general tendency seen in the figures is that a lower friction result in a faster response and opposite for larger friction. Both the NMPC and eMPC is seen to be robust to change in friction, as the step response with the three different friction values settles at the same value. The settling value of the PI controller is not the same when friction is changed. A better tracking result is actually seen with decreased friction, which could be due to a fortunate combination between the static friction and the quantised force output.

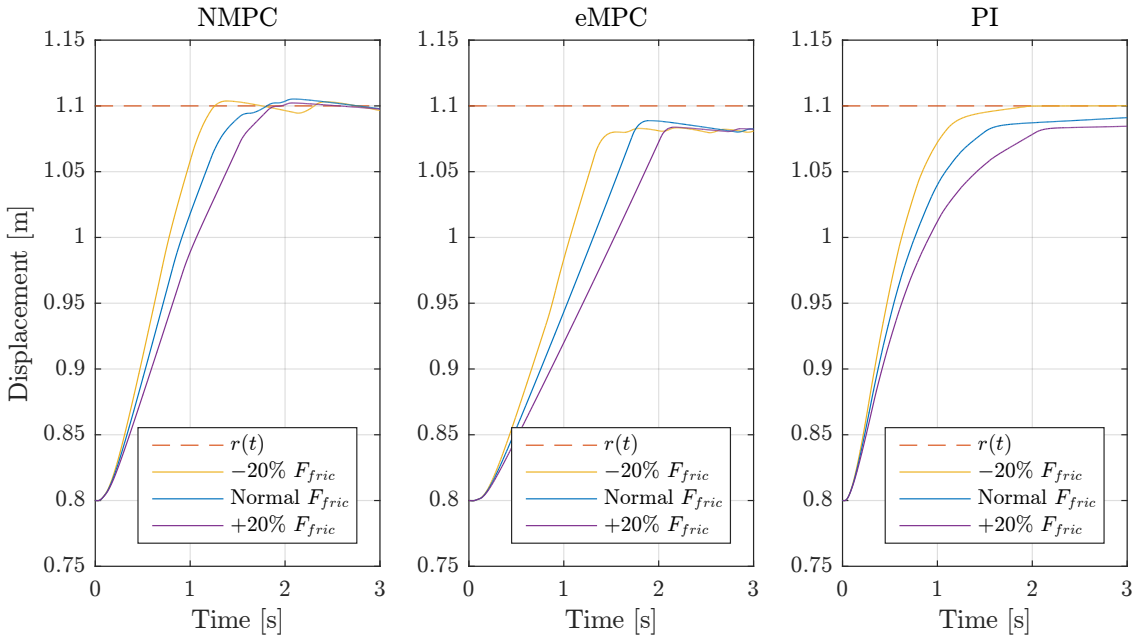


Figure 12.9. Position step response for NMPC, eMPC and PI position control for change in friction.

In figure 12.10 the position step response is shown for change in line pressure, which directly affects the available force. All three controllers are seen to be robust to change in line pressure. The first part of the response is slightly different for NMPC and eMPC, but without any significant difference. The largest difference is seen for the eMPC. However, the position almost settles at the same value for the three different values of line pressures.

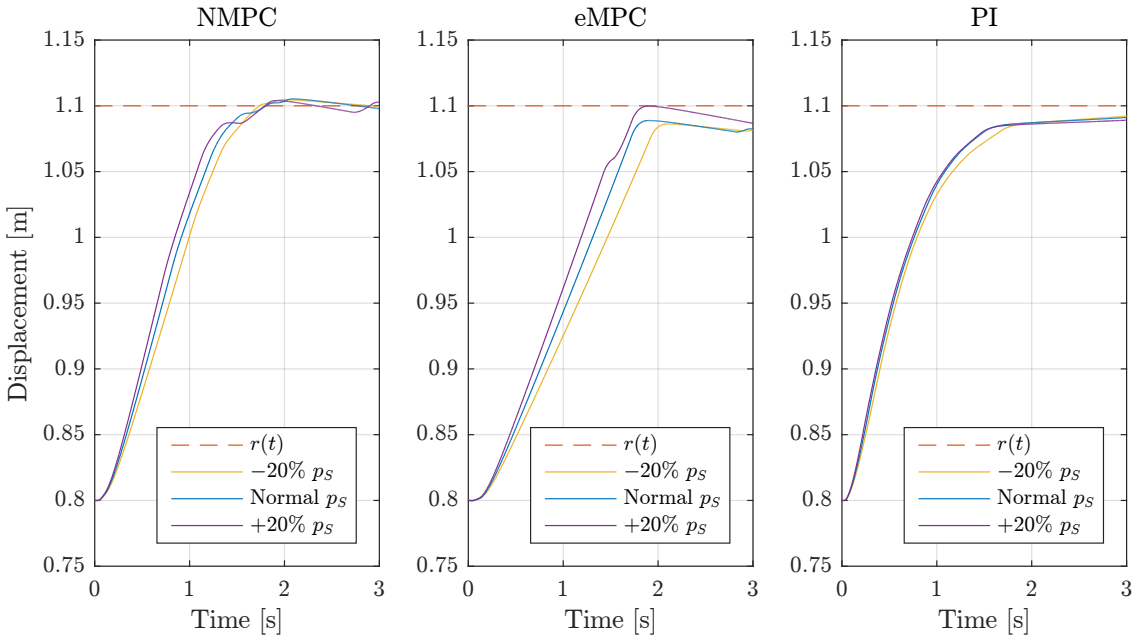


Figure 12.10. Position step response for NMPC, eMPC and PI position control for change in line pressure.

In figure 12.11 a velocity step response is shown for all three velocity controllers with change in friction. Both NMPC and eMPC are seen to be sensitive to change in friction, especially a smaller friction affects the step response. The PI controller show good response for an increased friction, but an offset is seen when the friction is decreased. As the friction directly affects the velocity, it is evident that a changed friction affect the velocity response. Generally the NMPC and eMPC is not seen to be very robust to change in friction, while the PI controller show some robustness to change in friction.

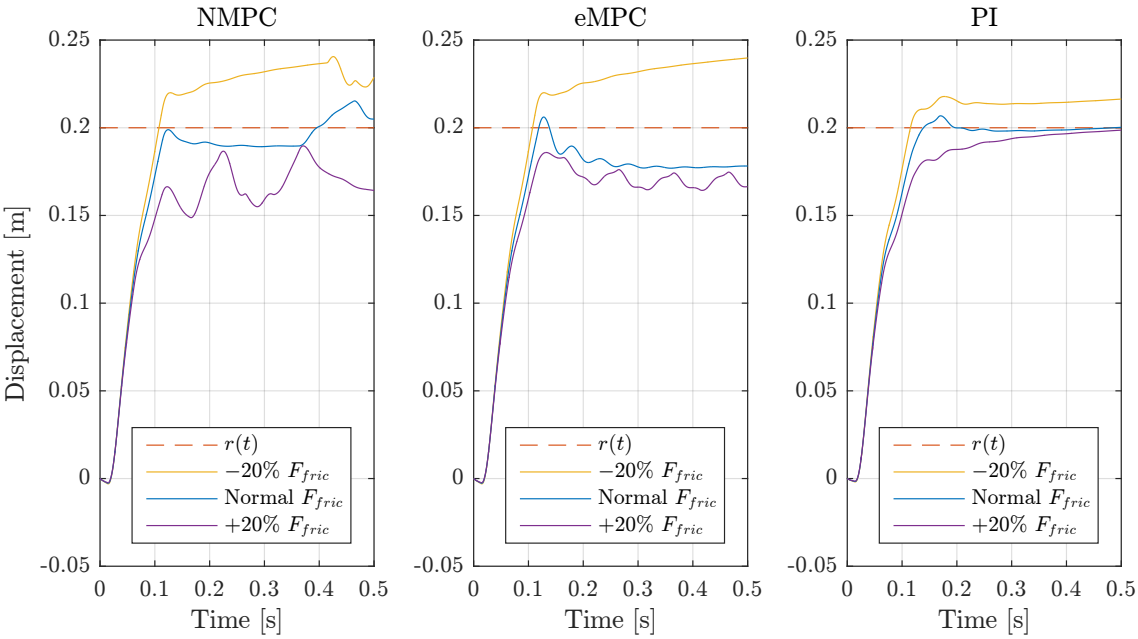


Figure 12.11. Velocity step response for NMPC, eMPC and PI velocity control for change in friction.

Figure 12.12 show the velocity step response for all controllers for different line pressure values. The settling value of the NMPC is very dependent on the line pressure values. In contrast the eMPC and PI controller show some robustness to change in line pressures. The eMCP response has an offset on both sides of the reference dependent on the change in line pressure. For increased line pressure the PI controller response is seen to reach what could look like a limit-cycle around the reference. As shown in section 11.3 the developed AW gain is determined based on the control gain, which is determined based on the system gain. Thus, by increasing the line pressures, the calculated AW gain no longer 'ignore' the quantisation. With the increased line pressures the AW gain is to small. Hence, limit-cycles can occur. Generally change in line pressures are seen to affect the velocity response of all controllers. However, the eMPC and PI velocity controller shown some robustness to change in line pressure.

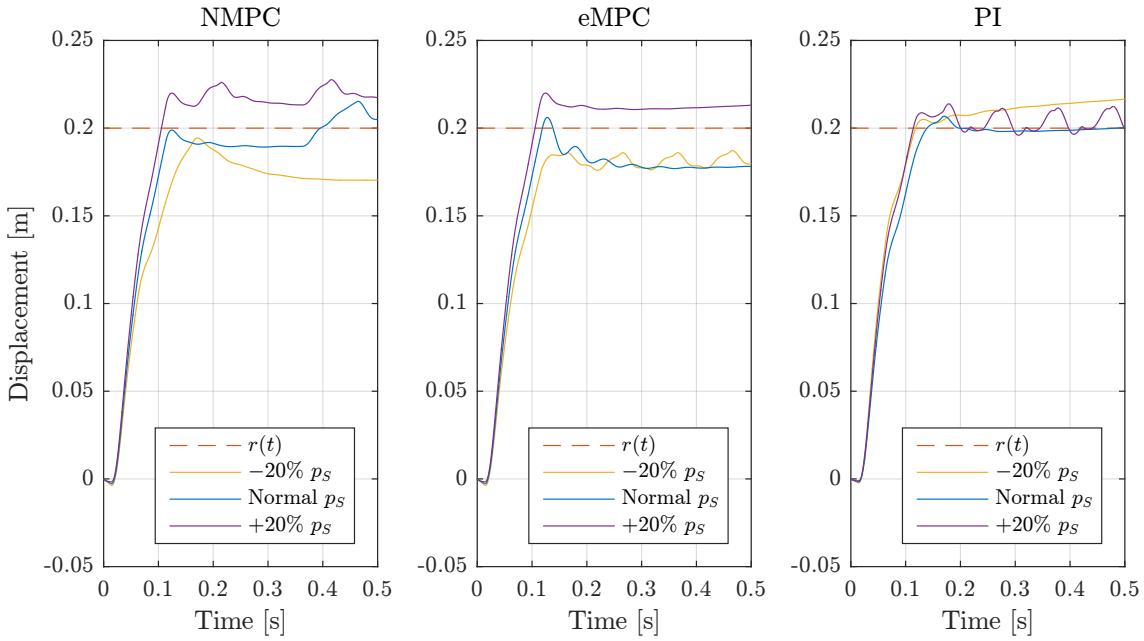


Figure 12.12. Velocity step response for NMPC, eMPC and PI velocity control for change in line pressure.

12.4 Chapter Summary

The NMPC, eMPC and PI controller has in this chapter been compared. The position controllers has been compared for a step response and load holding capabilities. For the compared step, the three controllers show similar position response. However, the NMPC and eMPC controller reduces the switching losses with 72.2% and 89.4% respectively compared to the switching loss with PI control. For load holding the switching losses are similar for all controllers, even though the PI controller switches significantly less. Generally the comparison of the position controllers showed that it was possible to achieve low position error, while considering the losses associated with switching. The velocity controllers has been compared for a step response and a sine reference. Similar RMS-error was seen for the three controllers for the velocity step, while the switching losses with eMPC was reduced by 68% compared to PI control. For the sine reference the eMPC was seen to reduce both RMS-error and switching losses by 19.9% and 66.7% respectively, compared with PI control.
In the robustness analysis the performance of the three controllers was evaluated for changes in friction and line pressures. All position controllers showed robustness to change in both friction and

line pressures. The velocity controllers generally showed a smaller degree of robustness to similar changes in friction and line pressures, with the PI controller showing most robustness.

CHAPTER 13

CONCLUSION

The use of digital hydraulics have received renewed interest due to potentially large energy savings, particularly for large scale applications such as in the off-shore industry. A state of the art analysis was conducted to investigate the current technologies and advances within digital hydraulics, and three main topologies were described: switch control, digital flow control units and secondary control. This master's thesis set to investigate control strategies for secondary control of a discrete displacement cylinder (DDC) as a test setup is readily available at Aalborg University.

The discussion following the state of the art analysis concluded that, while many secondary control investigations seeks an optimum between tracking performance and energy efficiency, none of the suggested implementations could claim optimality. Hence, a hypothesis of this thesis was to investigate if such an optimal control strategy for secondary controlled discrete displacement cylinder was possible. It was found in [5], that MPC had been used to control a switch controlled pneumatic clutch application with discrete inputs. The authors had developed an explicit MPC solution for the problem, based on solving the NMPC problem for several points in a confined state space.

Part I

In order to develop model based controllers a detailed mathematical model of the discrete displacement cylinder test setup was made. This included modelling transmission line dynamics using momentum and continuity equations. The valve manifold was modelled using the orifice equation and cylinder chamber pressure dynamics was described with continuity equations. Uncertain parameters such as friction was experimentally determined.

The model was validated by comparison to experimental results and considered sufficiently accurate for control development. Furthermore, the energy losses associated to switching between pressure levels was described as well as the throttling losses based on the work in [12].

Part II

Analysis of the mathematical model revealed the possibility of limit cycles due to the discrete nature of the system, and further analysis revealed a relation between physical parameters of the system and the amplitude and frequency of the limit cycle. Particularly, a small load mass and large pressure line

differences was revealed to have a negative impact on the size and frequency of the limit cycles. A fundamental difficulty of load holding was also described, caused by a mismatch between available discrete force levels and the required force.

The test bench at the university showed to be impractical for position and velocity control as the mass of the load was too small. Furthermore, no real life application case study was readily available. Hence, an imagined application case was synthesised to be able to test the control strategies developed. The focus of the application case was to replicate difficult control problems found in real life applications more than describing a real life application.

NMPC Controller

A NMPC controller was developed to form a basis for the later work on creating an explicit approximation of the NMPC solution. A discrete time reduced order model was described, where pressure dynamics where neglected. A cost function penalising tracking error and switching losses was synthesised and physically motivated constraints where formulated. The NMPC problem was solved using the *glcFast* routine of the TOMLAB optimisation package, which yields a close-to-global solution to the problem.

Results showed that it is possible penalise switching losses while maintaining good tracking performance for both position and velocity control. However, it was furthermore concluded that the NMPC is not implementable in real-time due to the computational heavy optimisation problem.

eMPC Controller

An eMPC control was developed in order to achieve the results obtained by the NMPC control in real time. The eMPC control was developed based on the work in [5]. The complete algorithm to convert the NMPC problem to explicit form was described and implemented in MATLAB and several modifications where added to increase the speed of convergence for the algorithm. Due to a very large required computational effort, the problem size was reduced in dimension and size and convergence parameters where increased.

The eMPC solutions based on the NMPC problem showed to have good tracking performance and be able to reduce switching costs for both position and velocity control. Some steady state errors was seen, and they are assumed to be due to the large approximation tolerances.

Linear Control

For comparison, a linear position and a linear velocity control was developed. Anti wind-up was investigated with the purpose of reducing or completely removing limit cycle behaviour. Anti wind-up developed for the velocity control problem was showed to be able to remove limit cycles at the cost of introducing a small steady state error. No such solution was found for the linear position control.

Part III

A comparison between eMPC, NMPC and the linear control showed that eMPC and NMPC was able to reduce the switching losses significantly compared to the linear control, while having equal or better tracking performance. For a step response in position, the eMPC control had a 35% larger RMS-error compared to the PI control, but switching losses where reduced by 89%. For a velocity step response,

the eMPC and PI control had similar RMS-error, but the eMPC reduced switching losses by 68%. For a sine wave trajectory, the eMPC velocity control achieved 20% smaller RMS-error and 67% less switching losses than the linear equivalent.

An analysis of robustness showed that all three controllers showed robustness for changes in friction and pressure supply level for position control. For velocity control, all three controllers showed a smaller degree of robustness for similar changes, with a small advantage to the linear control.

Finally, it can be concluded that position and velocity control is possible for a secondary controlled DDC. While the tracking error may not be on par with a proportional valve controlled system, energy efficiency are greatly reduced, which makes it an interesting alternative. It has been shown that switching costs can be further reduced by applying optimal control in the form of eMPC. The computational time required to find an explicit approximate solution makes this control best suited to integrated solutions, such as was seen in NorrDigiTM. However, as the number of pressure levels and cylinder chambers increase, so does the number of discrete dimensions of the eMPC problem, increasing the required computational effort and reducing the usability of the eMPC algorithm. Hence, it is best suited for applications with a small amount of discrete force levels.

13.1 Future Work

The developed eMPC controllers was based on large approximation tolerances and a reduced state space and it was considered reasonable that performance differences between the eMPC and the NMPC could be reduced by reducing the approximation tolerances and solving the problem for the full state space. It then remains to be shown that this is indeed the case, and thus the implementation of the algorithm should be further developed to be able to support parallel computing for faster problem solving.

The control developed in this thesis was evaluated by simulation only. It is important for the proof of concept of eMPC on a DDC that it is tested on a real life application or an appropriately scaled test bench.

Large pressure oscillations was seen in the simulated results on the full order model, due to the long transmission lines between the valve manifold and the DDC chambers. In order to reduce the pressure oscillations future work could be to investigate and implement valve opening trajectories.

The test setup available at Aalborg University was seen to be impractical for position and velocity control testing on the DDC. However, during this thesis, a lot of work have been put in developing pressure control for the load side cylinder. It is encouraged that this work is continued. This includes validating the control performance of the developed pressure control as well as work on emulating various high mass loads on the DDC by developing pressure trajectories based on state feedback.

BIBLIOGRAPHY

[1] M. Linjama, "Digital fluid power - state of the art," *The Twelfth Scandinavian International Conference on Fluid Power*, 2011.

[2] M. Linjama, H.-P. Vihtanen, A. Sipola, and M. Vilenius, "Secondary controlled multi-chamber hydraulic cylinder," *The 11th Scandinavian International Conference on Fluid Power*, 2009.

[3] R. H. Hansen, *Design and Control of the PowerTake-Off System for a Wave Energy Converter with Multiple Absorbers*. PhD thesis, Aalborg University, 2013.

[4] H. Kogler, R. Scheidl, M. Ehrentraut, E. Guglielmino, C. Semini, and D. G. Caldwell, "A compact hydraulic switching converter for robotic applications," *Fluid Power and Motion Control*, 2010.

[5] A. Grancharova and T. A. Johansen, "Explicit approximate model predictive control of contstrained nonlinear systems with quantized input," *Int. Workshop on Assessment and Future Directions of NMPC*, 2008.

[6] A. Plöckinger, M. Huova, and R. Scheidl, "Simulation and experimental results of pwm control for digital hydraulics," *The Fifth Workshop on Digital Fluid Power*, 2012.

[7] H. Sande, T. A. Johansen, G.-O. Kaasa, S. R. Snare, and C. Bratli, "Switched backstepping control of an electropneumatic clutch actuator using on/off valves," *Proceedings of the 2007 American Control Conference*, 2007.

[8] A. H. Hansen, *Investigation and Optimisation of a Discrete Fluid Power PTO-system for Wave Energy*. PhD thesis, Aalborg University, 2014.

[9] A. Laamanen, M. Linjama, and M. Vilenius, "On the pressure peak minimization in digital hydraulics," *The Tenth Scandinavian International Conference on Fluid Power*, 2007.

[10] M. Huova, A. Laamanen, and M. Linjama, "Energy efficiency of three-chamber cylinder with digital valve system," *International Journal of Fluid Power*, 2014.

[11] M. Huova and A. Laamanen, "Control of three-chamber cylinder with digital valve system," *The Second Workshop on Digital Fluid Power*, 2009.

[12] R. H. Hansen, T. O. Andersen, and H. C. Pedersen, "Analysis of discrete pressure level systems for wave energy converters," *International Conference on Fluid Power and Mechatronics*, 2011.

[13] D. R. Yemm, *Pelamis WEC - Full-Scale Joint System Test*, 2003.

[14] R. H. Hansen, T. O. Andersen, H. C. Pedersen, and A. H. Hansen, "Control of a 420 kn discrete displacement cylinder drive for the wavestar wave energy converter," *Proceedings of the ASME/BATH 2014 Symposium on Fluid Power and Motion Control*, 2014.

[15] A. H. Hansen, H. C. Pedersen, and M. M. Bech, "Avoidance of transmission line pressure oscillations in discrete hydraulic systems – by shaping of valve opening characteristics," *The Seventh Workshop on Digital Fluid Power*, 2015.

[16] A. H. Hansen and H. C. Pedersen, "Reducing pressure oscillations in discrete fluid power systems," *Journal of Systems and Control Engineering*, 2015.

[17] A. Dell'Amico, M. Carlsson, E. Norlin, , and M. Sethson, "Investigation of a digital hydraulic actuation system on an excavator arm," *The 13th Scandinavian International Conference on Fluid Power*, 2013.

[18] A. Sipola, J. Mäkitalo, and J. Hautamäki, "The product called norrdigi tm," *The Fifth Workshop on Digital Fluid Power*, 2010.

[19] A. Grancharova and T. A. Johansen, *Explicit Nonlinear Model Predictive Control - Theory and Applications*. Springer, 2012.

[20] M. AB, "Modeling of hydraulic systems, tutorial for the hydraulics library," 2013.

[21] Bucher Hydraulics, *DigiValve NS 10 NC-WS22GDA-10*, 2014. Datasheet.

[22] T. O. Andersen, *Fluid Power Systems, Modelling and Analysis*. Aalborg University, 2003.

[23] T. O. Andersen and M. R. Hansen, *Fluid Power Circuits, System Design and Analysis*. Aalborg University, 2007.

[24] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.

[25] J. b. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[26] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*. Wiley International, 2004.

[27] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica 36, 789-814*, 1999.

[28] T. Glad and L. Ljung, *Control Theory: Multivariable and Nonlinear Methods*. Taylor and Francis, 2000.

[29] A. Grancharova and T. A. Johansen, "Approximate explicit model predictive control incorporating heuristics," *I: Proceedings. IEEE International Symposium on Computer Aided Control System Design*, 2002.

[30] D. R. Jones, "Direct global optimization algorithm," *Encyclopedia of Optimization, vol. 1, pp. 431–440*, 2000.

Part IV

Appendices

APPENDIX A

EXPERIMENTAL DESCRIPTION

A.1 Determination of Unknown Constants

To determine the friction coefficients of the non-linear friction model a simple experiment have been conducted. The friction model is described in section 5.3, and is here expressed with all viscous, Coulomb and Stribeck -friction. A standard velocity/friction relation is shown in figure A.1 and it follows equation A.1.

$$F_{fric} = \tanh\left(\frac{\dot{x}_s}{\gamma}\right)\left(F_{C1} + F_{C2}e^{-|\dot{x}_s|/k_{stri}}\right) + B_{vs}\dot{x}_s \tag{A.1}$$

where γ is a coefficient used to control the slope of the tangent hyperbolic function around $\dot{x}_s = 0$, F_{C1} is the Coulomb friction constant and F_{C2} and k_{stri} are coefficients related to the stiction-type Stribeck friction. B_{vs} is the viscous friction coefficient.

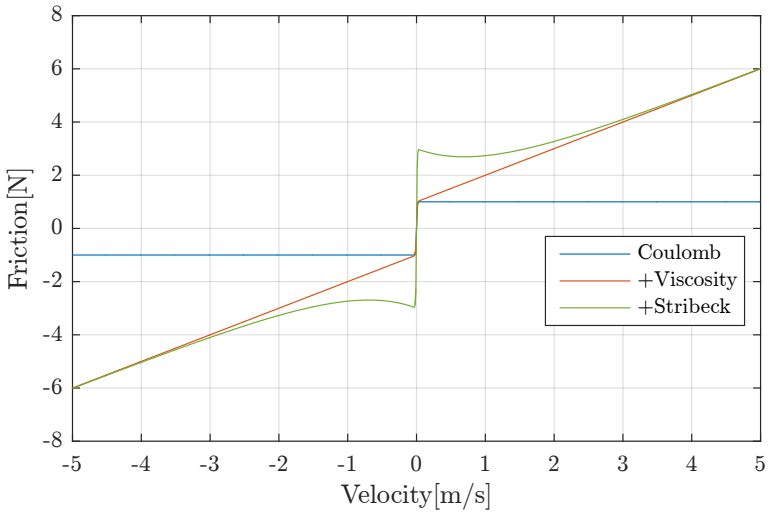


Figure A.1. Coulomb friction model with and without viscosity and Stribeck effects

To investigate the velocity/friction relation of the test-setup used in this thesis the loadside cylinder is operated to run with sinusoidal position symmetric around the centre position. This is done by giving a low frequency sinusoidal position reference to the loadside position control. For the low

changing reference the position control is able to accurately control the position. By doing this the cylinder velocity will also follow a sinusoidal tendency and the system will thereby experience a range of velocities around zero. When the cylinder velocity crosses zero the Coulomb and Stribeck effects are clarified. An example of the position and velocity references are seen on figure A.2.

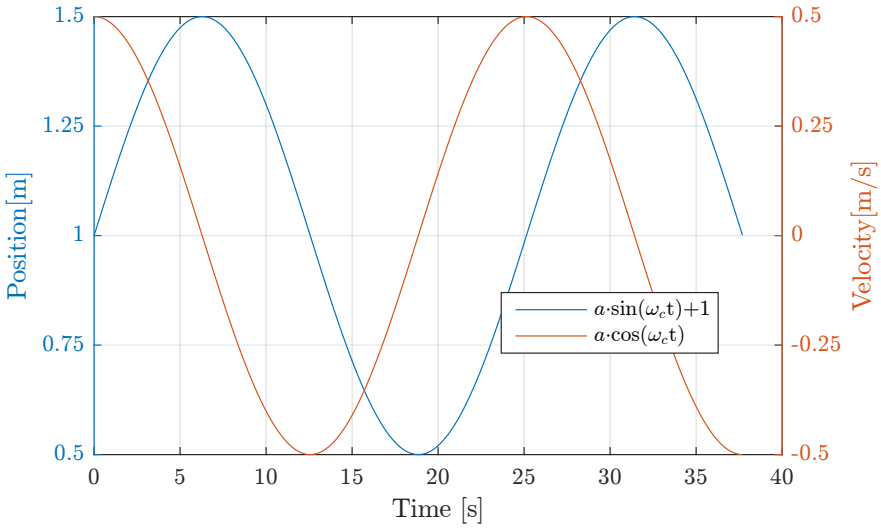


Figure A.2. Example of position and velocity reference with: $\omega_c = 0.25 \frac{rad}{s}$ and $a = 0.5$

Since the is not possible to directly measure the friction directly it will be estimated by measuring some of the system variable. The estimated friction is then found by rewriting Newtons 2. law from section 5.3 on page 27 to equation A.2.

$$F_{fric}(\dot{x}_s) = F_{DDC} - F_{LS} - m_s \ddot{x}_s \tag{A.2}$$

A.1.1 Experimental Results

Three experiments have been conducted to determine the system friction at different velocity ranges. The DDC-cylinder is for these test set at a constant force level. The force level with the lowest force is chosen. Then a velocity controller is used to track the velocity reference. There are no acceleration sensors on the test setup, so therefore the acceleration is estimated by differentiating the velocity reference.

First is the sinusoidal velocity reference given in a range of [-0.5:0.5] $\frac{m}{s}$, in order to determine friction at larger velocities. The result from this is seen in figure A.3.

Next two experiments are made in lower velocity ranges of [-0.1:0.1] $\frac{m}{s}$ and [-0.15:0.15] $\frac{m}{s}$. From these experiments it is easier to extract the Coulomb and Stribeck effects. Figures A.4 and A.5, shows the results from these experiments.

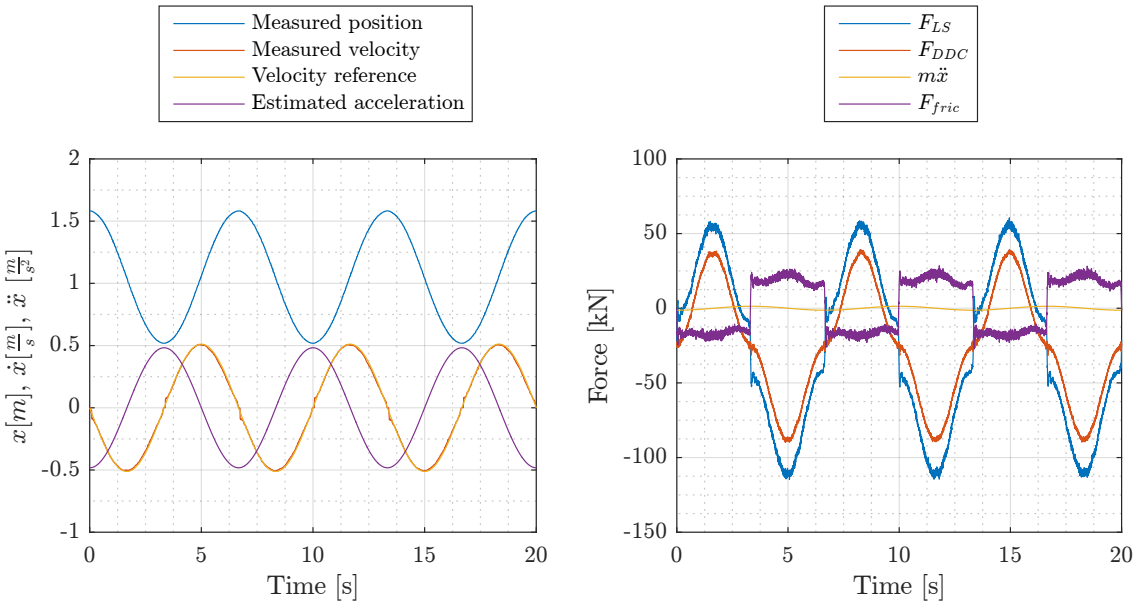


Figure A.3. Test 1: High velocity range [-0.5:0.5] $\frac{m}{s}$.

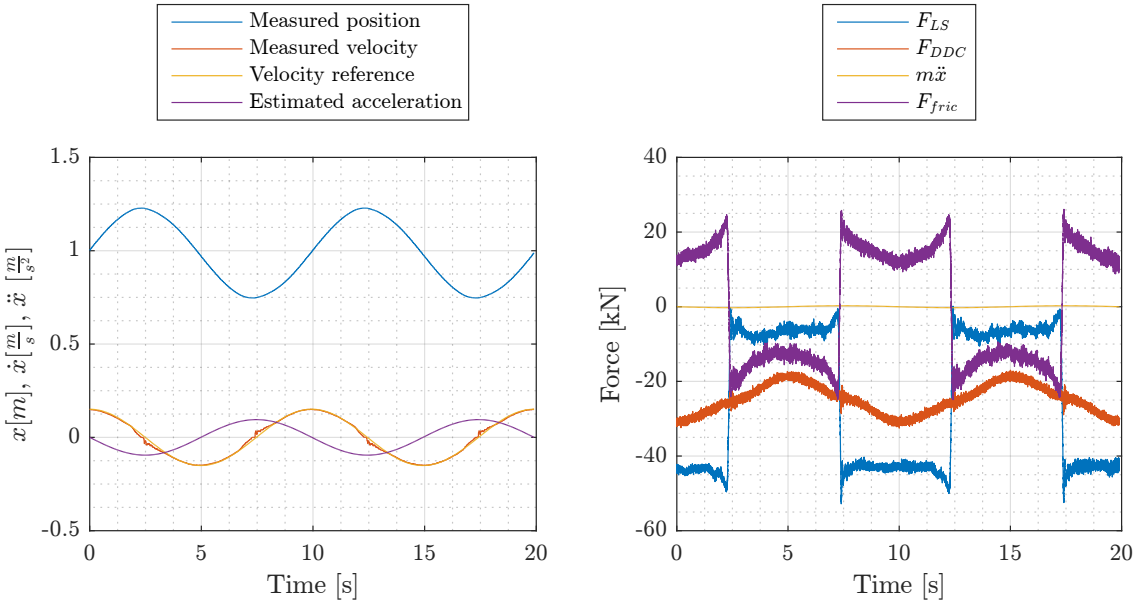


Figure A.4. Test 2: Low velocity range [-0.15:0.15] $\frac{m}{s}$.

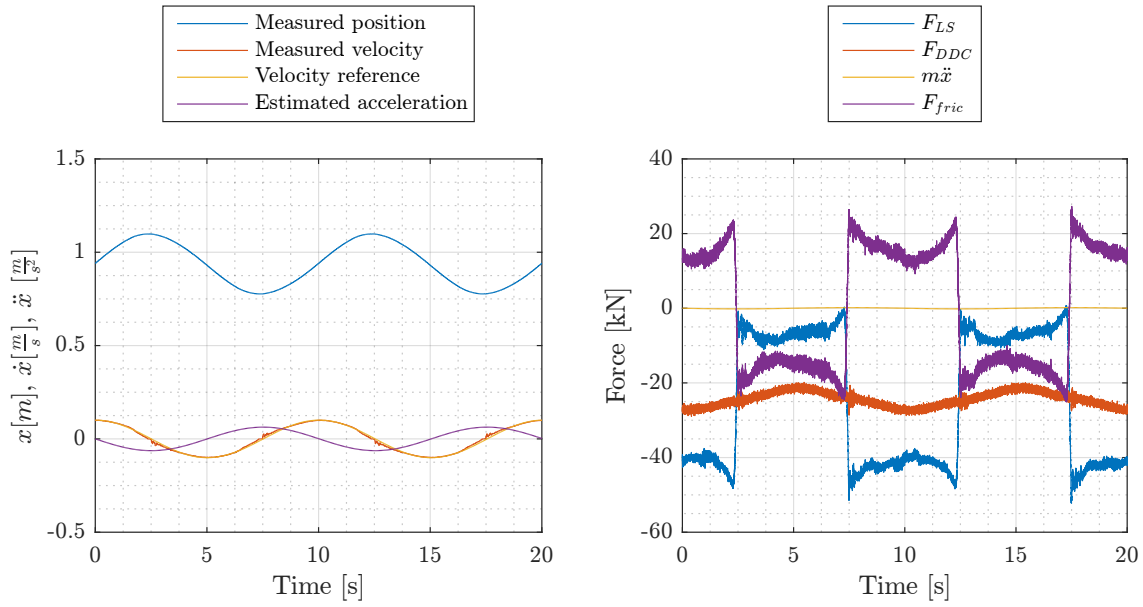


Figure A.5. Test 3: Low velocity range [-0.1:0.1] $\frac{m}{s}$.

The friction forces from all three experiments are plotted in a velocity/friction plot, seen on figure A.6. Together with the experimental data is a optimal and a fitted analytic expression for friction with the parameters shown in table A.1.

Implementation of the friction in Simulink, showed some uncontrolled oscillation in the numerical solver caused by the decreasing nature of the Stribeck friction and the lack of damping. In order to make the dynamics of the Simulink model work as seen in real life a compromise have been made on the friction model. The revisited friction curve is illustrated with the dashed line on figure A.6.

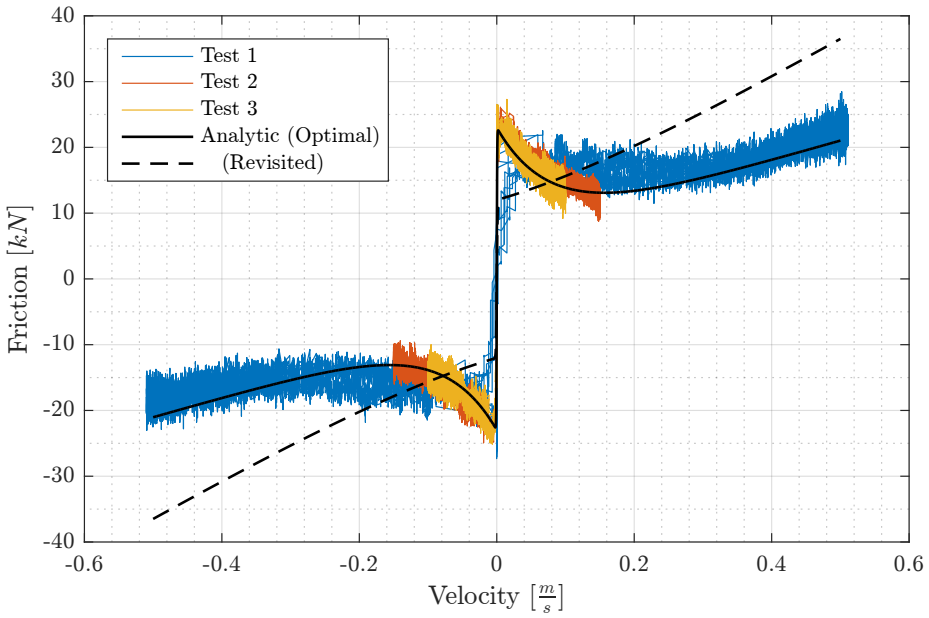


Figure A.6. Velocity-friction plot.

Name	Variable	Optimal fit	Revisited
Coulomb friction constant	$F_{C1}[N]$	6000	6000
Stribeck friction constant	$F_{C2}[N]$	17000	6000
Stribeck time constant	k_{stri} $[m/s]$	0.08	0.2
Slope of tangent hyperbolic function	γ [-]	$6 \cdot 10^{-4}$	$1 \cdot 10^{-3}$
Viscous friction coefficient	B_{vs} $[Ns/m]$	30000	60000

Table A.1. Friction parameters

APPENDIX B

NUMERICAL OPTIMISATION

The routine "glcFast" of the TOMLAB optimisation environment in MATLAB is used to solve the optimum of the objective function. It implements an extended version of the DIRECT optimisation algorithm presented in [30]. The DIRECT algorithm as presented in [30] is only able to handle inequality constraints, but the algorithm used in the routine "glcFast" is modified to be able to handle equality constraints as well. In the following the general concept of the DIRECT algorithm will be presented.

The DIRECT algorithm (DIviding RECTangles) is a direct search method and therefore not based on the gradient of the objective function or an approximate of it, i.e. it is a derivative-free method. An advantage of the algorithm is that it can handle problems that may contain both continuous and integer variables, and the functions may be nonlinear and non-smooth.
The general problem solved by the DIRECT algorithm can be written as

$$V^*(\mathbf{x_0}) = \min_{\mathbf{z}} f(\mathbf{z}, \mathbf{x_0}) \tag{B.1}$$

subject to:

$$g_j(\mathbf{z}, \mathbf{x_0}) \leq 0 \quad , \quad j = 1, 2, \dots m \tag{B.2}$$

$$l_i \leq z_i \leq u_i \quad , \quad i = 1, 2, \dots r \tag{B.3}$$

$$z_i \in \mathbb{I} \quad , \quad i = r+1, r+2, \dots s \tag{B.4}$$

$$\tag{B.5}$$

where $\mathbb{I}$ is the set of variables that are restricted to integer values. The vector of optimisation variables $z = [1, \dots z_r, z_{r+1}, \dots, z_s]$, includes both variables $(1, \dots, z_r)$ and integer variables $(z_{r+1}, \dots, z_s)$. The bounds on the variables limit the search to a s-dimensional hyper-rectangle. The algorithm proceeds by dividing the hyper-rectangle into smaller rectangles, where the function value is sampled at each center. In figure B.1 the first three iterations of the algorithm for a hypothetical problem with two optimisation variables are shown.

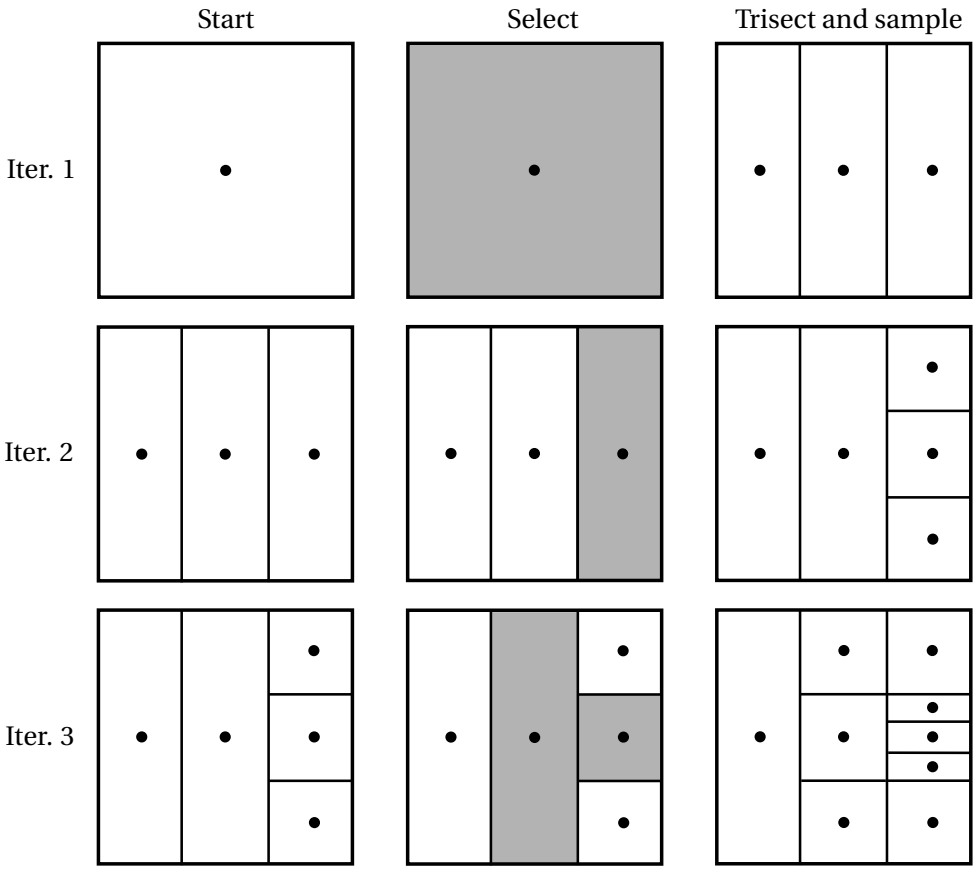


Figure B.1. Partition of the hyper-rectangle with the DIRECT algorithm [30].

For each iteration the partition of rectangles is evaluated and one or more rectangles is selected for further search. The selected rectangles are then trisected along one of its long sides and the center points of the new rectangles are sampled. For the first iteration the entire space is covered by one rectangle and the selection of rectangles is therefore trivial. For the second iteration three rectangles can be chosen and in this example one is selected. For the third iteration 5 rectangles can be selected and two is selected in this example.
The key concept of the algorithm is the selection of rectangles, since it determines how the entire space is evaluated. The rectangles are selected based on weightings of local versus global search. The difference between global and local search can be explained by considering the two extreme cases. If a pure global search strategy were used one of the biggest rectangles would be selected every iteration and the rectangles would become small at the same rate. By the end, the sample points would form a uniform grid. With the global strategy every part of the search space is considered and optimum parts is not overlooked. On the other hand, if only local search strategy were used, the rectangle with the best objective function value would be selected every time, which would result in a fast solution. However, an optimum could be overlooked, if it was found in a rectangle with a poor objective function value at the center. Before presenting how the algorithm selects the rectangles it will be explained how inequality constraints are treated.

The algorithm uses an auxiliary function that combines information about the objective and constraint functions. A notation denoting the center point of the p'th rectangle is introduced as z_p. Furthermore, positive weighting coefficients for the inequality constraints are introduced as $\varphi_1, ..., \varphi_m$. The weighting coefficients are calculated based on the ratio between the average rate

of change of the objective function and the constraints. Denoting the global optimum as V^* and the minimum objective function value at the current iteration as $V_{min}(\mathbf{x_0})$, V^* is assumed to satisfy $V^* \leq V_{min}(\mathbf{x_0}) - \epsilon$, where ϵ influence the accuracy of the solution. Values for ϵ can be found in [30]. The auxiliary function, evaluated at the center of rectangle p, is then

$$V_p^a(V^*, \mathbf{x_0}) = \max\{f(z_p, \mathbf{x_0}) - V^*, 0\} + \sum_{j=1}^{m} \varphi_j \max\{g_j(z_p), 0\} \tag{B.6}$$

The first term penalise objective function values above V^*. The second term is a sum of weighted constraint violations. The lowest possible value of the auxiliary function is ϵ and occurs at the global minimum. At other points the auxiliary function is larger either due to infeasibility or objective function values larger than the optimum.

The selection of rectangles is based on the rate of change of the auxiliary function. In order to evaluate this, a measure of the rectangles sizes is introduced as d_p and this expresses the distance from centre point to vertices of the p'th rectangle. The rate of change of the auxiliary function for the p'th rectangle can then be expressed as

$$h_p(V^*, \mathbf{x_0}) = \frac{\max\{f(z_p, \mathbf{x_0}) - V^*, 0\} + \sum_{j=1}^{m} \varphi_j \max\{g_j(z_p), 0\}}{d_p} \tag{B.7}$$

The selection algorithm is based on the rate of change function h_p as function of V^*. An example with 5 rectangles is shown in figure B.2.

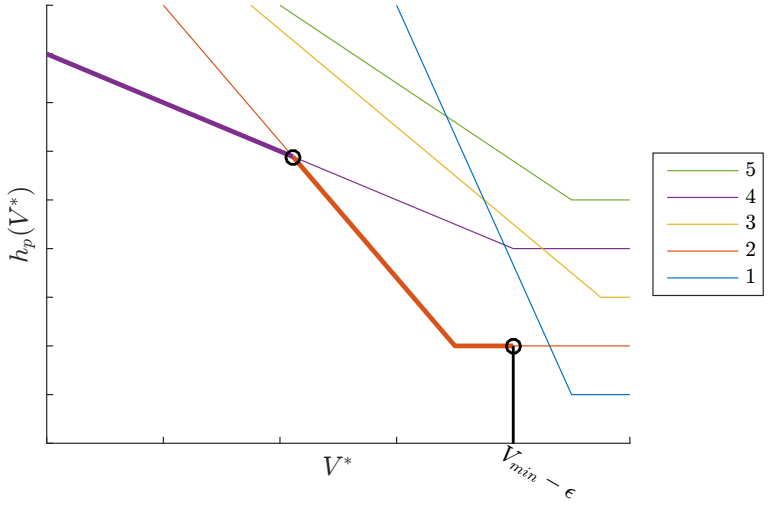


Figure B.2. Rate of change as function of V^* used for selecting rectangles.

For $V^* > V_{min}(\mathbf{x_0}) - \epsilon$ the first part of the rate of change function is zero and thus the function is constant due to the constant constraint value for each rectangle. For $V^* < V_{min}(\mathbf{x_0}) - \epsilon$ the rate of change function increases. The rectangles in the lower left envelope represents the rectangles that are selected by the algorithm. For the example shown in the figure rectangle 2 and 4 would be selected. The selected rectangles are then trisected and the algorithm is ready for a new iteration. Integer variables are handled with minor changes in the trisection routine and the way the center point of the rectangle is defined. The center point of a rectangle consisting of integers is defined as the floor of the algebraic average. For a rectangle consisting of two integer variables both with the range [1:8],

the center point is (4,4). The rectangles containing integer variables, can not necessary be trisected such that the new rectangles have the same size. However, the trisection is performed such that the center point of the trisected rectangle, remains the center of one of the new rectangles. In order for the algorithm to work best, the integer variables has to be ordered, e.g the integer variable with the range [1:8] is ordered from 1 to 8. The integer variables are ordered so the function value in a rectangle's center can be expected to be indicative for what the function is like in the rest of the rectangle.
The stopping criteria of the algorithm is the number of function evaluations, which is determined by the user. However, the algorithm implemented in "glcFast" is modified so an evaluation time of the algorithm can be used as stopping criteria as well. The stopping criteria does not insure that the calculated optimum is actually the optimum of the objective function. Thus, the function evaluations or evaluation time of the algorithm has to be considered in order to obtain the required accuracy of the solution.

In [30] disadvantages and advantages of the algorithm are stated. A disadvantage of the space-partitioning approach is that relatively tight lower and upper bounds on all variables are required for the algorithm to work well. Due to the space-partitioning approach the algorithm works best for low-dimensional problems, which is stated as less than 20 variables. The algorithm is efficient in terms of the number of function evaluations required to get close to the global minimum, but takes longer to achieve a high degree of accuracy. Thus, the authors suggest that the best performance of the algorithm is obtained by combining the DIRECT algorithm with a good local optimiser. The algorithm works well when small uncertainties or noise is present, because it result in a small change in objective function values, which usually has little impact on the set of selected rectangles until late in the search. The main advantage of the algorithm is its unique approach to balance local and global search, by the simple idea of not sampling just one point per iteration, but rather sampling several points using all possible weightings of local versus global search.

A step-by-step procedure of the DIRECT algorithm is found in [30].

APPENDIX C

EVALUATION OF COMPUTATIONAL PARAMETERS

C.1 Function Evaluations

To determine the number of function evaluations that should be used by the numerical optimiser, the solution to the cost function is evaluated as function of number of function evaluations. The cost function is dependent on the state vector $\mathbf{x}(t)$. Thus, the solution to the cost function is evaluated for three different initial values of $\mathbf{x}(t)$. The cost function values as function of the number of function evaluations are shown in figure C.1 for a step where the initial value of the position is larger than the reference, where the initial position is less than the reference and where the initial position is the same as the reference. For all three cases the initial velocity is zero. It should be noted that the optimum determined by the numerical optimiser is not necessarily determined on the last function evaluation, but could be determined on a previous function evaluation.

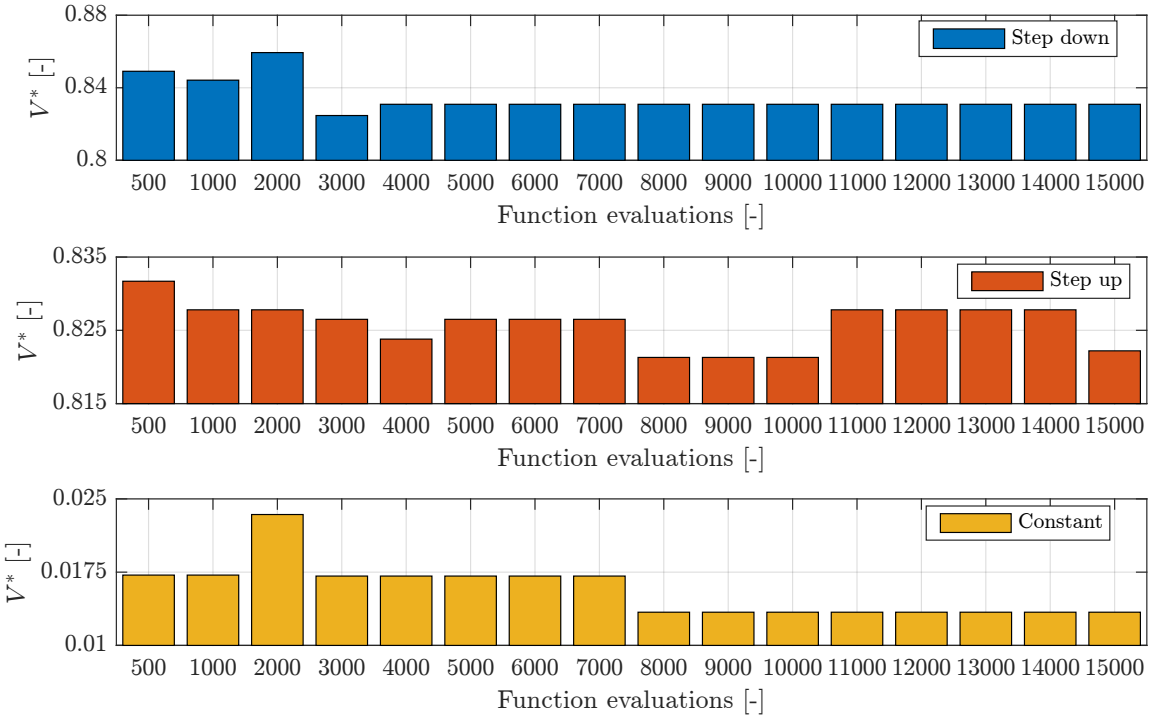


Figure C.1. Cost function values as function of the number of function evaluations for three different initial values of $mathbfxt$

For some cases it is seen that the cost function value increases for an increase in the number of function evaluation, this is because the cost function values are calculated based on the actual output at each time steps. Thus, the horizon for each step is not considered in the cost function value. A difference in the first step in the horizon, which is the actual output, can therefore change the optimisation problem in the next time step and thereby change the overall cost function value.
As can be seen from the figure, it is not possible to determine a clear number of function evaluations, for which the cost function values are at a constant minimum for all three cases. However, the tendency is that for number of function evaluations less than 3000 an increase in cost function is seen. Based on the cost function values shown in the figure 4000 function evaluations is chosen for the numerical optimiser. It is possible that the accuracy of the solution could be increased with a larger number of function evaluations, but it would at the same time result in an increased computational time of the numerical optimiser.

C.2 Computational Time

On average the computational time is mainly divided between the *glcFast* optimisation routine, evaluation of state equations 9.2, 9.3, evaluation of the input equation 9.6 and evaluation of cost function equation 9.14. Figure C.2 show the relative computational time for each sub-function of an optimisation step for 500, 3000 and 15000 function evaluations.

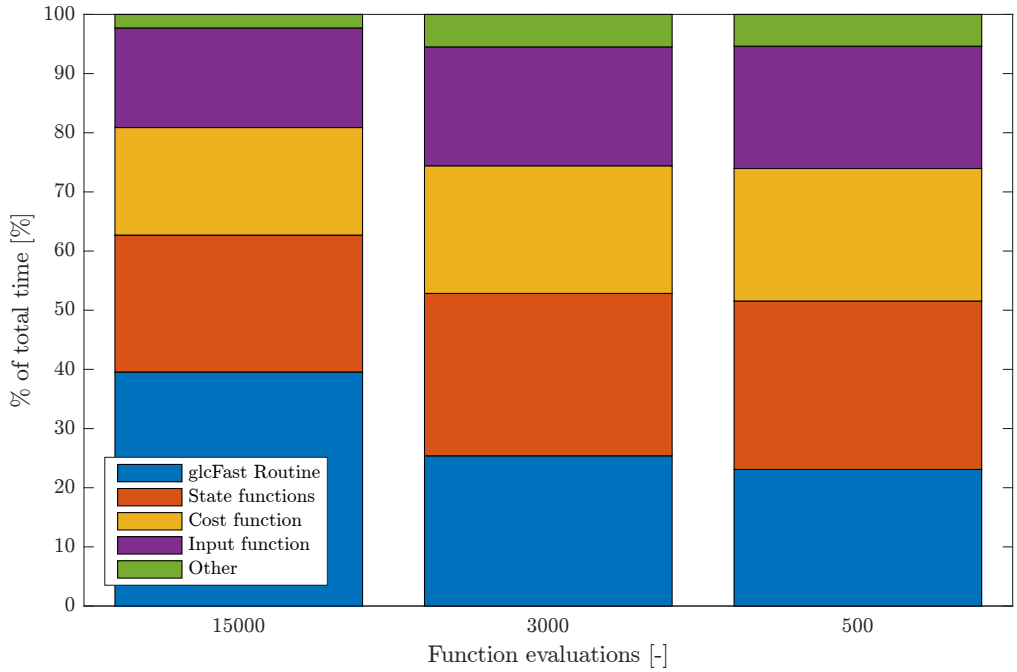


Figure C.2. Computational time of sub-functions in an optimisation step.

The average computational time for each time step was 0.23 seconds, 1.3 seconds and 9 seconds for 500, 3000 and 15000 function evaluations respectively. It can be seen, that a great part of the computational time is spent with user supplied functions. Hence, those function should be optimized with respect to computational time. MATLAB code for the user supplied function in C.2 are shown in appendix D, sections D.3, D.4 and D.5 for the discrete system equations, cost function equation and input equation respectively.

APPENDIX D

ALGORITHM MATLAB CODE

D.1 Main loop code

The MATLAB implementation of the eMPC algorithm is shown here. Each step is mainly calculated in separate functions, and hence, the code shown here illustrates the book-keeping and code-flow of the implementation.

```
run model_script;

%% Initialise
clear xdata wdata subidx W
global optParam mpcParam sysParam

% Set MPC parameters
mpcParam.N = 15;    %Set mpc horizon.
mpcParam.Ts = 0.1;  %Set Euler forward integration time.
mpcParam.N0 = 0;    %Set number of sub-HR to be constructed in X0.
mpcParam.Q = 1*diag([15,0.1]); %State cost (x1,x2).
mpcParam.R = 1*diag([1,0.0005]); %Input cost (dU, U).
mpcParam.P = 1*mpcParam.Q; %Terminal state cost (x1,x2).


% Set system parameters
sysParam.x1c_U = S.L;    % Upper constraint on y.
sysParam.x1c_L = 0;      % Lower constraint on y.
e1c_TU = 0.7;   % Upper terminal constraint on x1.
e1c_TL = -0.7;   % Lower terminal constraint on x1.
x2c_U = 0.4;    % Upper constraint on x2.
x2c_L = -0.4;   % Lower constraint on x2.
e2c_TU = 0.4;   % Upper terminal constraint on x2.
e2c_TL = -0.4;   % Lower terminal constraint on x2.

%Set constraint vectors for horizon N, incl. terminal constraint
c_Ux1vec = [sysParam.x1c_U*ones(mpcParam.N,1);e1c_TU];
```

```
28 c_Ux2vec = [x2c_U*ones(mpcParam.N,1);e2c_TU];
29 c_Lx1vec = [sysParam.x1c_L*ones(mpcParam.N,1);e1c_TL];
30 c_Lx2vec = [x2c_L*ones(mpcParam.N,1);e2c_TL];
31
32 %Set limits of state space
33 sysParam.x_U = [0.3, 0.4, S.L-0.75];  % Upper limits [x1_U, x2_U,...,
      xN_U]
34 sysParam.x_L = [-0.3, -0.4, 0.75];    % Lower limits [x1_L, x2_L,...,
      xN_L]
35
36 %Load data structs
37 sysParam.DDC = DDC;          % Load DDC struct from DDC parameters.
38 sysParam.S = S;              % Load S struct from sleigh parameters.
39 sysParam.A = A;              % Load A struct from application
      parameters.
40 sysParam.G = G;              % Load G struct from sleigh parameters.
41
42 %Set bounds for input U
43 sysParam.u_U = 3*ones(3*mpcParam.N,1)';   % Upper bounds for U.
44 sysParam.u_L = ones(3*mpcParam.N,1)';     % Lower bounds for U.
45
46 %Set constraint vectors
47 N1 = 1+(1+mpcParam.N0)*(2^length(sysParam.x_U)+2*length(sysParam.x_U)
      );
48 sysParam.c_U =  [c_Ux1vec;c_Ux2vec]; % Upper bounds for constraints.
49 sysParam.c_L =  [c_Lx1vec;c_Lx2vec];   % Lower bounds for constraints
50 sysParam.c_U_hat = [repmat(c_Ux1vec,[N1 1]);repmat(c_Ux2vec,[N1 1])];
51 sysParam.c_L_hat = [repmat(c_Lx1vec,[N1 1]);repmat(c_Lx2vec,[N1 1])];
52
53 % Setting U to be integer variables
54 sysParam.IntVars = 1:length(sysParam.u_L);
55
56 % Initial input vector
57 sysParam.U0 =[2,2,2];
58
59 % Set optimiser parameters
60 optParam.maxFcnEval = 4000;  %Set maximum function evaluations.
61 optParam.maxApproxFcnEval = 4000;  %Set maximum function evaluations.
62 optParam.MaxCPU = [];        %Set maximum CPU time (in seconds).
63 optParam.prntLvl = 0;   %Set print level: 0 none, 1 some, 2 all.
64
65 %Cost function for step 4.
66 optParam.costFcn = 'glcOpt_f_v3'; %Name of cost function.
67 optParam.cnstrFcn = 'glcOpt_c_v3'; %Name of constraint function.
68
69 %Cost function for step 7.
70 optParam.costApproxFcn = 'glcApproxOpt_f_v3';
```

```
71  optParam.cnstrApproxFcn = 'glcApproxOpt_c_v3';
72
73  %Set up glcFast (some parameters change for step 7).
74  optParam.Prob = glcSetup_v3();
75
76  % Set algorithm parameters
77  epsVol = 0.01*prod(sysParam.x_U-sysParam.x_L); %Tolerance size of
        rectangle
78  expV = 0; %Initially explored volume
79  eps_a = 0.02;    %Absolute tolerance level of approximation error.
80  eps_r = 0.05;     %Relative tolerance level of approximation error.
81
82  % Create empty and zero-sets used in book keeping.
83  emptyNSet = cell(1,2^(size(sysParam.x_U,2)));
84  zeroNSet = cell(1,2^(size(sysParam.x_U,2)));
85  zeroNSet(:) = {0};
86  empty2Set = cell(1,2);
87  zero2Set = cell(1,2);
88  zero2Set(:) = {0};
89
90
91
92  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
93  %  Step 1: Initialise state space  %
94  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
95  display('Step 1: Initialising state space as hyper-rectangle.')
96  xdata.X{1} = [sysParam.x_U;sysParam.x_L];
97  % Define different sets used in book-keeping
98  xdata.feas{1} = [];
99  xdata.xplrd{1} = 0;
100 xdata.V_hat{1} = [];
101 xdata.u_hat{1} = [];
102 subidx = 0;
103
104 wdata.W = [];
105 wdata.V = [];
106 wdata.ExitFlag = [];
107 wdata.U = [];
108 startTime = tic;  %Start timer
109
110
111 % Plot initialisation (2D only)
112 p = patchIni(xdata.X); %Initialise partition plot.
113
114 % Save name for data
115 saveName = sprintf('%s%d%d%d', 'xdata_', sysParam.U0);
116 %% Start while-loop
```

```
117 while (1)
118     subidx = subidx+1; %Index to keep track of current hyper-
           rectangle
119 
120     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
121     %   Step 2: Select any unexplored hyper-rectangle  %
122     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
123 
124     currentTime = toc(startTime);
125     display('Step 2: Selecting unexplored hyper-rectangle.')
126     idx = find([xdata.xplrd{:}]==0,1);
127     explIdx = find([xdata.xplrd{:}]==1);
128 
129     % If no such hyper-rectangle exists, terminate.
130     if isempty(idx)
131         display('        No more unexplored hyper-rectangles.
               Terminating.')
132         fprintf('        Time since start: %0.0f minutes\n',
               currentTime/60)
133         load handel;
134         player = audioplayer(y, Fs);
135         play(player);   % Halleluja!
136         % Save partition data
137         save(saveName, 'xdata','wdata');
138         break
139     end
140     % Display information on current progress.
141     fprintf('        Time since start: %0.0i seconds\n',currentTime)
142     fprintf('        Number of state partitions: %0.0f\n', size(xdata
           .xplrd(:),1));
143     fprintf('        Volume of selected hyper-rectangle: %f\n',prod(
           abs(diff(xdata.X{idx},1,1))))
144     center = mean(xdata.X{idx});
145     fprintf('        Center of selected hyper-rectangle: (%0.3f,%0.3f
           ,%0.3f)\n',center(1),center(2),center(3))
146 
147     expV = 0;
148     if isempty(explIdx)
149         expV = 0;
150     else
151         for i = explIdx
152             expV = expV+prod(abs(diff(xdata.X{explIdx(i)},1,1)));
153         end
154     end
155     pctExpl = expV*100/prod(sysParam.x_U-sysParam.x_L);
156 
157     fprintf('        Volume explored: %0.2f pct.\n',pctExpl)
```

```
158
159     %Plot partition with color coding.
160     patchplot_v3(p,xdata.X,idx, xdata.feas);
161
162
163     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
164     %  Step 3: Generate a set of points W0 = {w0,w1,w2, ...,wN1 } %
165     %          associated to X0 by applying Procedure 1.1.        %
166     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
167     display('Step 3: Generating W0.')
168     W = step3_v3(xdata.X{idx});
169
170
171     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
172     %  Step 4: Compute a solution to the problem for x fixed to   %
173     %          each of the points wi, i = 0, 1, 2, ... , N1 by    %
174     %          using routine 'glcFast' of TOMLAB optimization     %
175     %          environment. If the problem has a feasible solution%
176     %          at all these points, go to step 7.                 %
177     %          Otherwise, go to step 5.                           %
178     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
179
180     fprintf('Step 4: Computing solutions for W0, set of %i points.\n'
           , size(W,2))
181     tic;
182     [wnew, wres] = step4_v3(wdata,W);
183     wdata.W{subidx} = wnew;
184     wdata.V{subidx} = ([wres(:).f_k]);
185     wdata.U{subidx} = ([wres(:).x_k]);
186     wdata.ExitFlag{subidx} = ([wres(:).ExitFlag]);
187     fprintf('        Solutions to W0 found in %0.1f seconds.\n', toc)
           ;
188
189     if ~isempty(find([wres(:).ExitFlag]~=0,1))
190
191
192         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
193         %  Step 5: Compute the size of X0 using some metric.   %
194         %          If it is smaller than some given tolerance, %
195         %          mark X0 infeasible and explored and go to   %
196         %          step 2. Otherwise, go to step 6.            %
197         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
198         volx = step5_v3(xdata.X{idx});
199         fprintf('Step 5: Volume of rectangle: %0.3f.\n',volx)
200         if volx<=epsVol
201             display('        Volume of X0 lower than tolerance,
               marking infeasible and explored and going to step 2.')
```

```
202             xdata.feas{idx} = 0;
203             xdata.xplrd{idx} = 1;
204             continue;
205         end
206
207
208
209         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
210         %  Step 6: If at least one of the points in wi,                   %
211         %           i = 0,1,2,...,N1 is feasible, split X0 into           %
212         %           hyper-rectangles X1, X2,..., XNs by applying          %
213         %           the Heuristic splitting rule 1.1. If none of          %
214         %           the points wi, i = 0,1,2,...,N1 are feasible,         %
215         %           split X0 into two hyper-rectangles X1 and X2 by       %
216         %           a hyperplane through its center point and             %
217         %           orthogonal to an arbitrary axis.                      %
218         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
219          xtmp = step6_v3(xdata.X{idx});
220          xdata.X(idx) = [];
221          xdata.feas(idx) = [];
222          xdata.xplrd(idx) = [];
223          xdata.V_hat(idx) = [];
224          xdata.u_hat(idx) = [];
225          xdata.X = [xdata.X, xtmp];
226          xdata.feas = [xdata.feas, emptyNSet];
227          xdata.xplrd = [xdata.xplrd, zeroNSet];
228          xdata.V_hat = [xdata.V_hat, emptyNSet];
229          xdata.u_hat = [xdata.u_hat, emptyNSet];
230          clear xtmp;
231          display('Step 6: Applying splitting rule and going to step 2.
                 ')
232          continue
233     end
234
235
236     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
237     % Step 7: Compute a constant function K0(x_hat) as an %
238     %         approximation to be used in X0.              %
239     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
240     display('Step 7: Computing a constant function K0 using Procedure
             5.1')
241     tic;
242     wdataApprox = step7_v3(wdata.W{subidx},wdata.V{subidx});
243     fprintf('        Constant solution to W0 found in %0.1f seconds.\
            n', toc);
244     if ~isempty(find([wdataApprox.ExitFlag]~=0,1))
245         % Find longest facet.
```

```
246         [~, splitIdx] = max(abs(diff(xdata.X{idx})));
247         % Split across longest facet .
248         xtmp = splitX_v3(xdata.X{idx}, splitIdx);
249         xdata.X(idx) = [];
250         xdata.feas(idx) = [];
251         xdata.xplrd(idx) = [];
252         xdata.V_hat(idx) = [];
253         xdata.u_hat(idx) = [];
254         xdata.X = [xdata.X, xtmp];
255         xdata.feas = [xdata.feas, empty2Set];
256         xdata.xplrd = [xdata.xplrd, zero2Set];
257         xdata.V_hat = [xdata.V_hat, empty2Set];
258         xdata.u_hat = [xdata.u_hat, empty2Set];
259         clear xtmp;
260         fprintf('        Infeasible K0, splitting X0 orthogonal to x%
               i into X1 and X2 and going to step 2.\n',splitIdx)
261         continue
262     end
263
264
265     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
266     % Step 8:  Compute an estimate e0_hat of the error bound e0 in %
267     %          X0 by applying Procedure 5.2.                        %
268     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
269     [eps_0_hat, splitIdx] = step8_v3(wdata.W{subidx},wdataApprox.x_k,
           wdata.V{subidx});
270     fprintf('Step 8: Estimate of error bound, ê0 of X: %f\n',
           eps_0_hat)
271     fprintf('        Minimum V(x): %f\n',min(wdata.V{subidx}))
272     eps_inc = 1;
273     %Increase tolerance if volume is below threshold (0.1% of total)
274     if prod(-diff(xdata.X{idx},1,1))<epsVol*0.1
275         eps_inc = 10;
276         fprintf('        Volume below threshold, increasing
               tolerances.\n')
277     end
278     if eps_0_hat<= max(eps_a*eps_inc,eps_inc*eps_r*min(wdata.V{subidx
           }))
279         xdata.feas{idx} = 1;
280         xdata.xplrd{idx} = 1;
281         xdata.V_hat{idx} = wdataApprox.f_k;
282         xdata.u_hat{idx} = wdataApprox.x_k;
283         display('        Accepting K0. Marking feasible and explored
               and going to step 2.')
284     else
285         % Split orthogonal to axis with greatest error gradient.
286         xtmp = splitX_v3(xdata.X{idx},splitIdx);
```

```
287         xdata.X(idx) = [];
288         xdata.feas(idx) = [];
289         xdata.xplrd(idx) = [];
290         xdata.V_hat(idx) = [];
291         xdata.u_hat(idx) = [];
292         xdata.X = [xdata.X, xtmp];
293         xdata.feas = [xdata.feas, empty2Set];
294         xdata.xplrd = [xdata.xplrd, zero2Set];
295         xdata.V_hat = [xdata.V_hat, empty2Set];
296         xdata.u_hat = [xdata.u_hat, empty2Set];
297         fprintf('        Rejecting K0. Splitting X0 orthogonal to x%i
                into X1 and X2 and going to step 2.\n',splitIdx)
298         clear xtmp;
299     end
300     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
301 end
```

D.2 Step 4

The code for **Step 4** is shown here. This code includes the steps taken to recognise if a point have previously been optimised in an earlier iteration.

```
function [W, Res] = step4_v3(wdata,W)
% Step 4, solve mp-MIP problem with glcSolve/glcFast for
% all points in W.
% Solver options are set in global struct optParam
% This implementation also recognises points already
% optimised in earlier iterations and excludes them
% from optimisation.
% Input:  - wdata, including previously optimised points
%         - subset W = [w1, w2,...wi]
% Output: - New subset of W, may be in a different order
%           due to reusing points
%         - Results from optimisation.

global optParam
Prob = optParam.Prob;
n = size(W(:),1); %n: number of points in W
m =  size(wdata.W,2); %m: number of sets W
if m>0;
    for i = 1:n
        for j = 1:m
            oldW = (cat(1,wdata.W{j}{:}));
            newW = W{i};
            % Index of oldW where rows of oldW are equal to newW(i).
            I{i} = find(ismember(oldW,newW,'rows')==1);
            if ~isempty(I{i})
                % i'th point in W found in j'th oldW in row I{i}.
                idxCell{i} = [I{i},j];
                % Do not search for more matches on point if already
                   found.
                break
            end
        end
    end
    for i = sort(find(~cellfun(@isempty,I)),'descend')
        W(i) = [];  %Delete those points that are already optimised
    end
end
l = size(W(:),1);
if l~=n   % If data has been deleted from W
    idxVector = cat(1,idxCell{:});
    fprintf('        W0 contains %d points from previous iterations.\
       n        Excluding points from optimisation and re-using
       results.\n',(n-l));
```

```
41 end
42 %Run optimiser for points in W.
43 for i = 1:l
44     Res(i) = glcOpt_v3(W{i},Prob);  %Get result struct,
45 end
46 if l~=n                            % If data has been deleted from W
47     for i = 1:size(idxVector,1)  % Add previously known data.
48         W{l+i} = wdata.W{idxVector(i,2)}{idxVector(i,1)};
49         Res(l+i).x_k = wdata.U{idxVector(i,2)}(:,idxVector(i,1));
50         Res(l+i).f_k = wdata.V{idxVector(i,2)}(idxVector(i,1));
51         Res(l+i).ExitFlag = wdata.ExitFlag{idxVector(i,2)}(idxVector(
               i,1));
52     end
53 end
```

D.3 Discrete Time System Function Code

Implementation of state equations 9.2, 9.3 in MATLAB. To increase optimisation speed only strictly necessary parameters are input to the function. Other system constants are defined in the function.

```
function x = sysFcn_v4(x0,p,dt)
% Discrete time implementation of the system.
% Explicit integration using forward Euler.
% Input:  - initial state x(t+k|t)
%         - input u(t+k|t)
%         - forward Euler timestep, dt
% Output: - x(t+k+1)

%Pre-allocate vectors (for speed)
x = zeros(size(x0'));

%Define constants
A_c1 = 0.0235; % Piston area 1 [m^2]
A_c2 = 0.0122; % Piston area 2 [m^2]
A_c3 = 0.0087; % Piston area 3+4 [m^2]
gamma = 1e-3; % Friction slope around 0 velocity [-]
Fc1 = 6000; % Coloumb friction [N]
Fc2 = 6000; % Stribeck friction [N]
k_stri = 0.2; % Stribeck coefficient [-]
Bv = 15*60e3; % Viscous friction [N/m/s]
M = 2000; % Mass [kg]
L = 8; % Length L [m]
g = 9.81; % Gravitational constant [m/s^2]
b = 2.236; % Length b [m]
e = 2; % Length e [m]
a = 2; % Length a [m]

% Discrete State Space form:
d = b*sqrt(abs(1-(((x0(1)+e)^2+b^2-a^2)/(2*(x0(1)+e)*b))^2)); %Moment
    arm
k = (a^2+b^2-(x0(1)+e)^2)/(2*a*b); %cos(theta)
Meq = L^2*M/d; %Equivalent mass [kg]

x(1) = x0(1)+dt*x0(2);
x(2) = x0(2)+dt*(-tanh(x0(2)/(gamma))*(Fc1+Fc2*exp(-abs(x0(2)/k_stri)
   ))-Bv*x0(2)+(-p(1)*A_c1+p(2)*A_c2+p(3)*(A_c3)-M*L*g*k/d))/Meq;
x(3) = x0(3);
end
```

D.4 Cost Function Code

Implementation of cost function equation 9.14 in MATLAB.

```
function V = costFcn_v3(x,u,Q,R,P)
% Cost function implementation.
% Input:  - state vector x(t+k|t) for k = 0,...,N
%         - input vector u(t+k|t) for k = 0,...,N-1
%         - weighting matrices Q, R, P
% Output: - cost function value V(x,u)

%Define constants
A_c1 = 0.0235; % Piston area 1 [m^2]
A_c2 = 0.0122; % Piston area 2 [m^2]
A_c3 = 0.0087; % Piston area 3+4 [m^2]
V_0c1 = 0.04814; % Initial chamber volume 1 [m^3]
V_0c2 = 0.00114; %I nitial chamber volume 2 [m^3]
V_0c3 = 0.002280; % Initial chamber volume 3+4 [m^3]
Fmax = 428120; % Maximum force [N]
max_loss = 2.7842598e4; % Maximum shift cost [J]
beta = 6000e5; % Bulk modulus [Pa]
Ls = 2; %Stroke length [m]
v_max = 0.5; %Maximum velocity [m/s]

% Normalise state vector
e = [(x(3,1)-x(1,:))/Ls; x(2,:)/v_max];

%Calculate state cost Q||x||^2+P||x||^2
stateCost = sum(sum(Q*(e(:,1:end-1).^2)))+sum(P*e(:,end).^2);

%Calculate input cost R||h(u,x)||^2
dU = diff(u,1,1)';  %Find pressure difference in time.
V_c = [-A_c1*x(1,1:end-1)+V_0c1; A_c2*x(1,1:end-1)+V_0c2; (A_c3)*x
    (1,1:end-1)+V_0c3]; %Find volumes
switchCost = sum(((sum((0.5*dU.^2).*(V_c/beta)))/(max_loss)).^2);

forceCost = sum(((-A_c1*u(1,2:end)+A_c2*u(2,2:end)+(A_c3)*u(3,2:end))
    /(Fmax)).^2);
inputCost = R(1,1)*switchCost+R(2,2)*forceCost;

V = stateCost+inputCost;
```

D.5 Input Function Code

Implementation of input function of equation 9.6 in MATLAB.

```
function p = inputFcn_v3(U)
%Input function, tranlates input constraints (1,2,3) to physical
   inputs to
%system.
%Input: input vector U = [u1, u2, u3]
%Output: pressure in chambers, p = [p1, p2, p3]

p_h = 200e5;  % High pressure supply level [Pa]
p_m = 107.5e5;  % Medium pressure supply level [Pa]
p_l = 20e5; % Low pressure supply level [Pa]


if U(1) == 1
    p1 = p_h;
elseif U(1) == 2;
    p1 = p_m;
else
    p1 = p_l;
end

if U(2) == 1
    p2 = p_h;
elseif U(2) == 2;
    p2 = p_m;
else
    p2 = p_l;
end

if U(3) == 1
    p3 = p_h;
elseif U(3) == 2;
    p3 = p_m;
else
    p3 = p_l;
end
p = [p1; p2; p3];
end
```

D.6 Step 8

Implementation of **Step 8** in MATLAB. This code includes approximating the error differences across possible splitting hyper-planes.

```
function [eps_0_hat, splitIdx] = step8_v3(W,K0,localV)
% Step 8 function, finds maximum error bound of K0.
% Input:  - subset W = [w1, w2,...wi]
%         - constant function K0
%         - cost function value set V^*(u^*,W)
% Output: - maximum error bound of K0
%         - Index value for largest error difference

global optValues optParam
Prob = optParam.Prob;
%Find V_hat for each point in W (is written in global variable)
for i = 1:size(W,2)
   W{i}(1) = W{i}(3)-W{i}(1);  %y(t) = r(t)-e(t)
end
Prob.user.W = W;
Prob.user.localV = localV;
[~] = glcApproxOpt_f_v3(K0,Prob);

n = size(localV,2);
eps_0 = zeros(n,1);
for i=1:1:n
    eps_0(i,1) = optValues.localApproxV(i)-localV(i);
end

for i = 1:size(W,2)
   W{i}(1) = W{i}(3)-W{i}(1);  %e(t) = r(t)-y(t)
end
% Project W to origo
Wtmp = cat(1,W{:})-repmat(W{1},n,1);
% Find error difference across first, second and third axis
dx = abs(sum(eps_0(Wtmp(:,1)>0))-sum(eps_0(Wtmp(:,1)<0)));
dy = abs(sum(eps_0(Wtmp(:,2)>0))-sum(eps_0(Wtmp(:,2)<0)));
dz = abs(sum(eps_0(Wtmp(:,3)>0))-sum(eps_0(Wtmp(:,3)<0)));

%Split across axis with largest error difference
[ ~, splitIdx] = max([dx dy]);

eps_0_hat = max(eps_0,[],1);
```

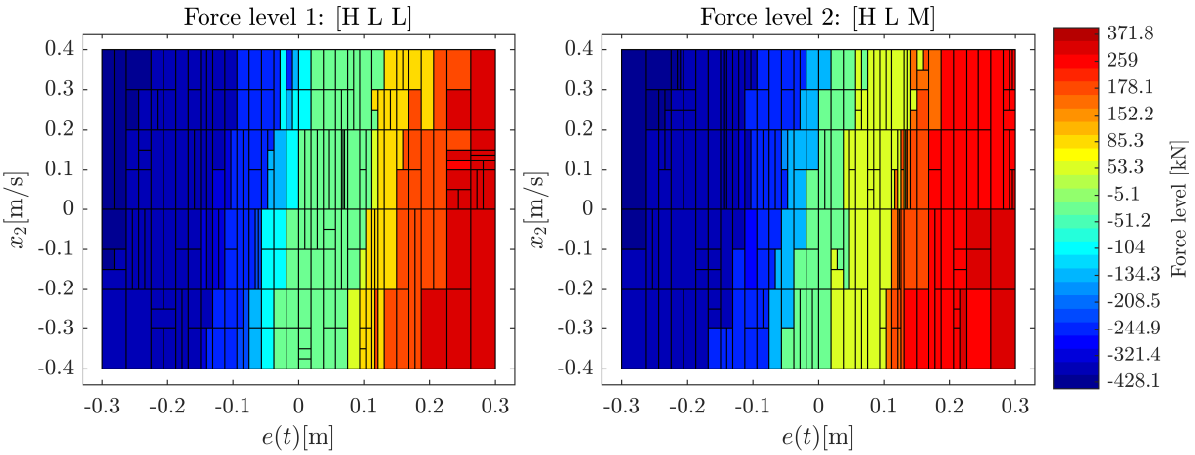
APPENDIX E

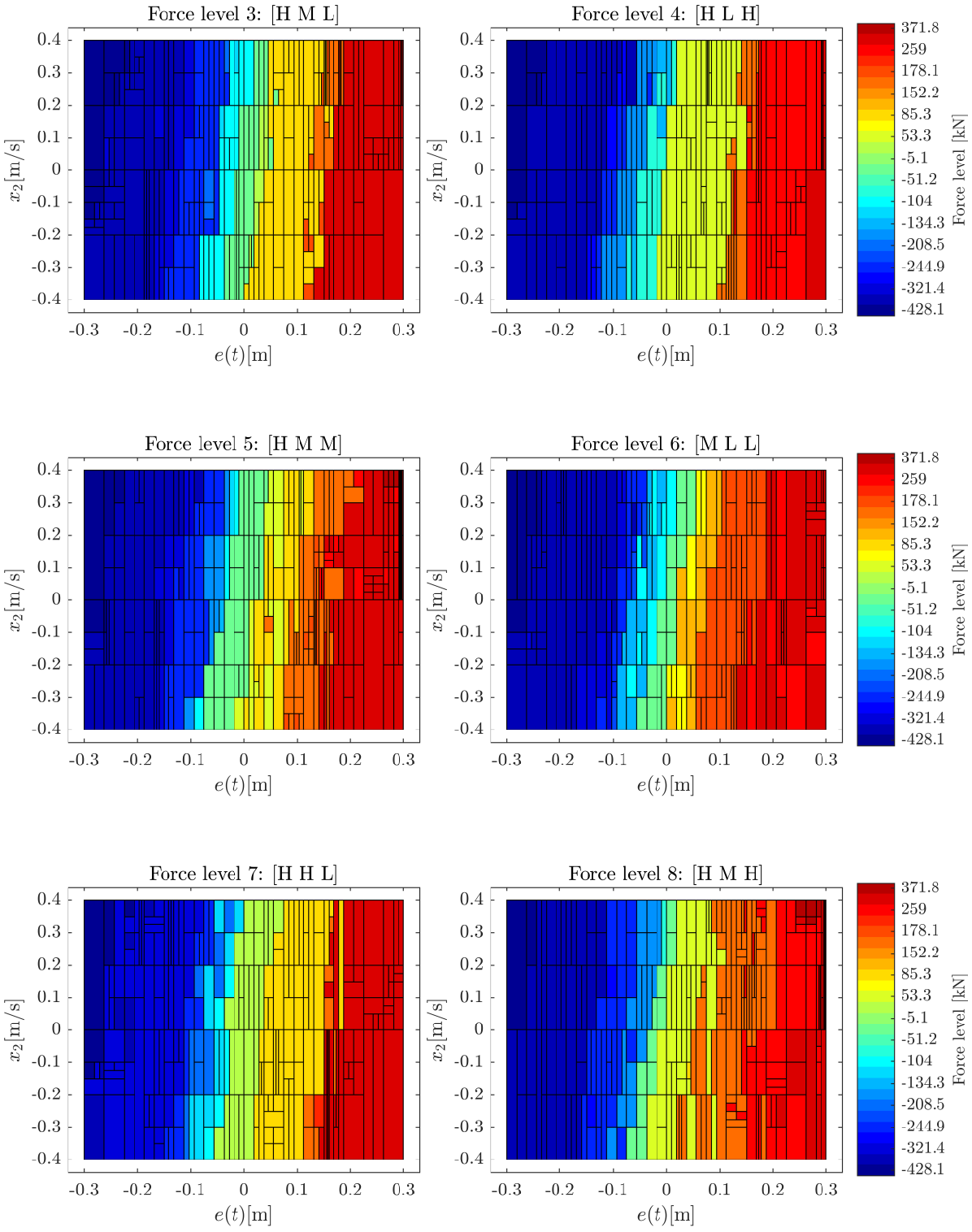
SOLUTION FOR EXPLICIT MODEL PREDICTIVE CONTROL

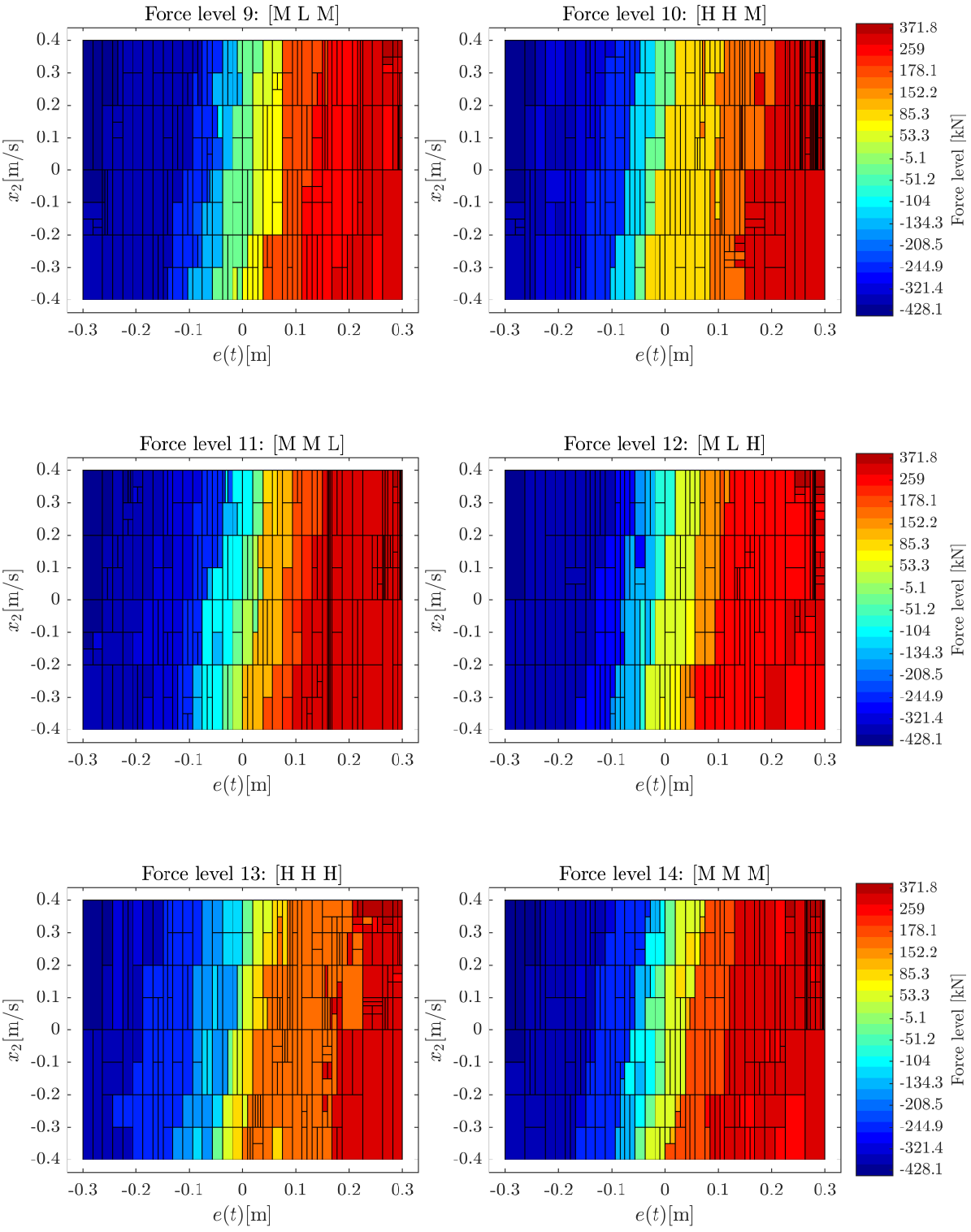
In this appendix the 27 hyper-rectangle solutions found with eMPC for both the position and velocity control are presented. First the solution for position control is presented, followed by solution for velocity control.

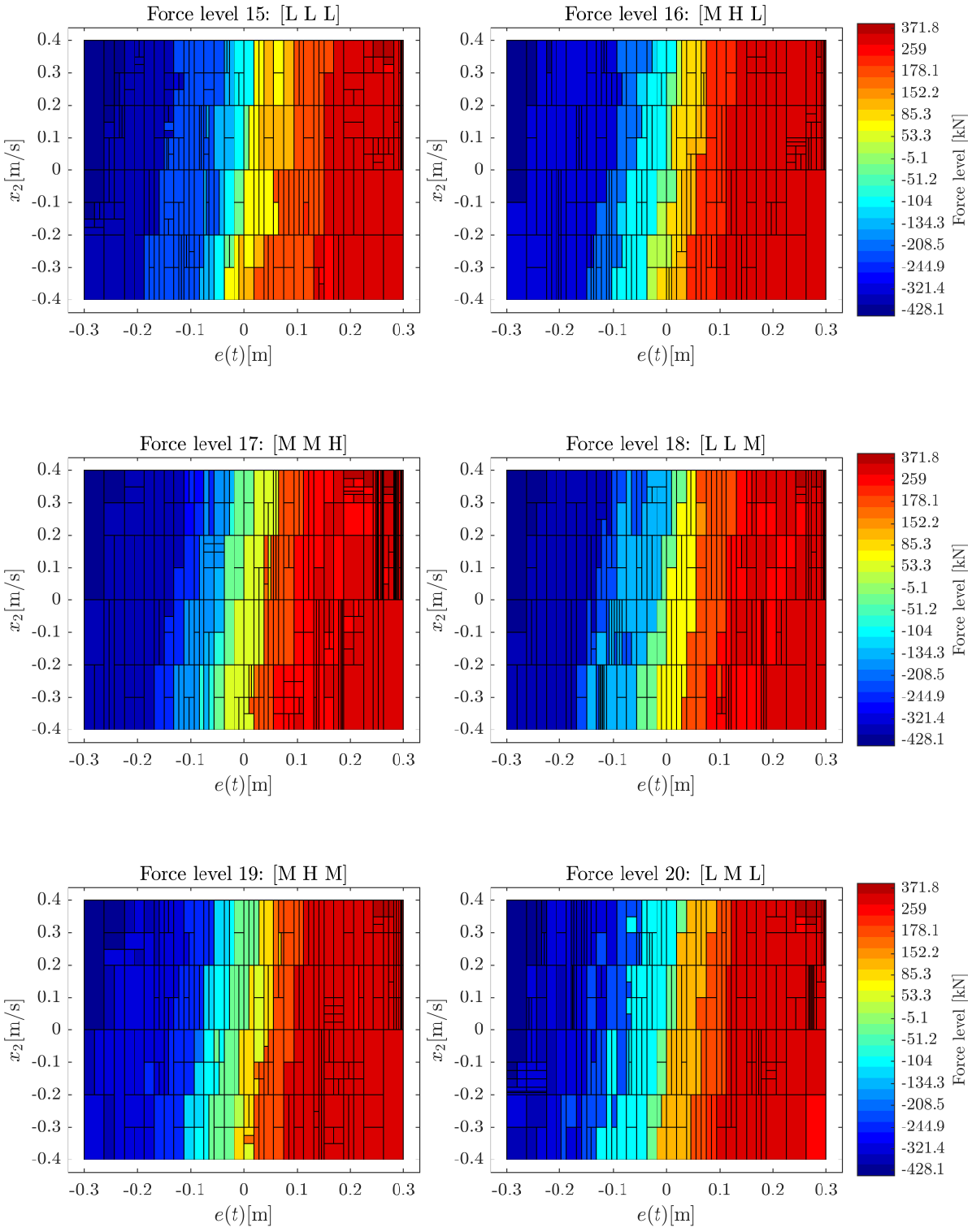
E.1 Solution for Explicit Model Predictive Position Control

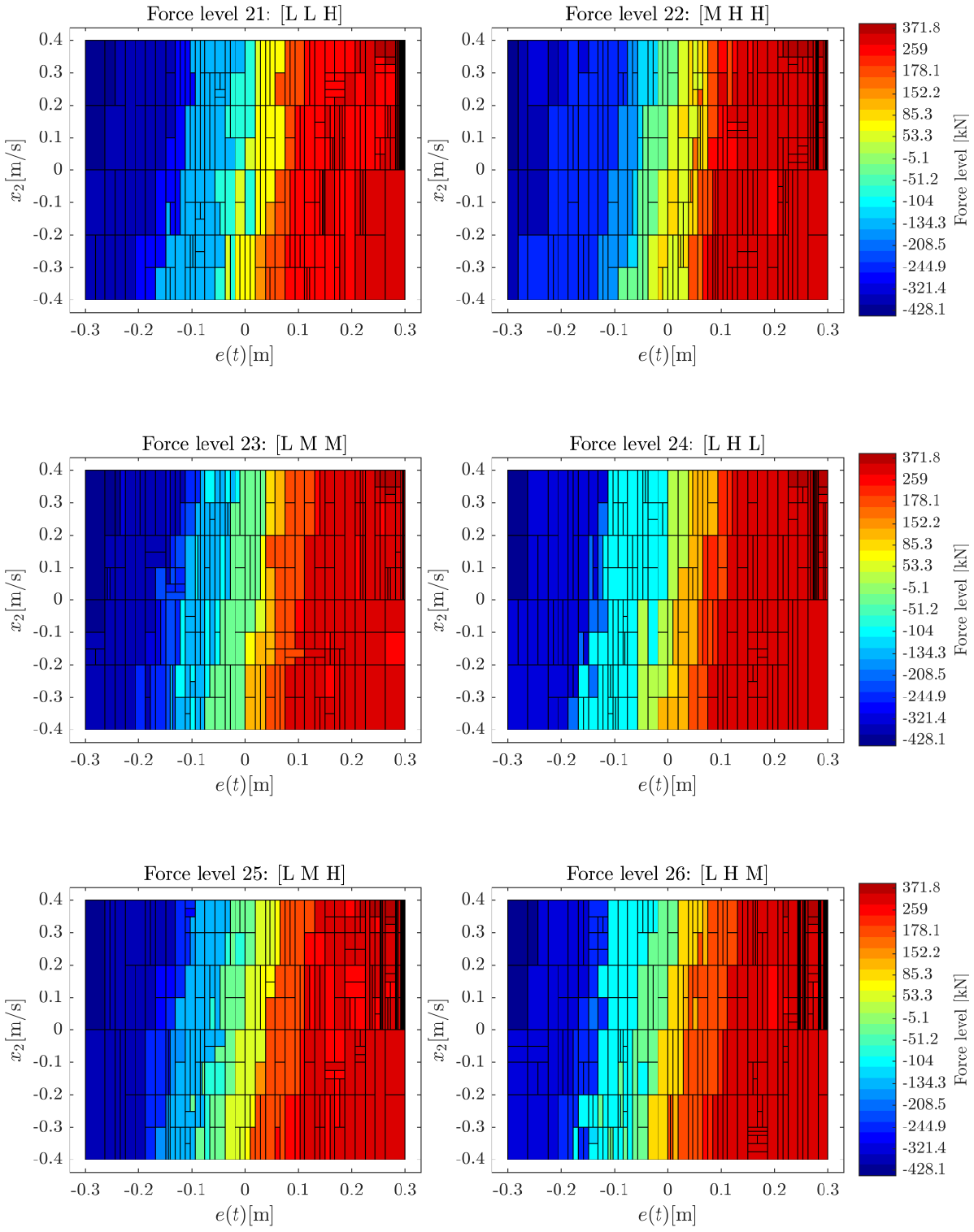
The explicite solution for the position control problem has the two continuous dimensions, which is the position error and the velocity. A solution is shown for each of the 27 force levels, which can be seen as the discrete dimension.

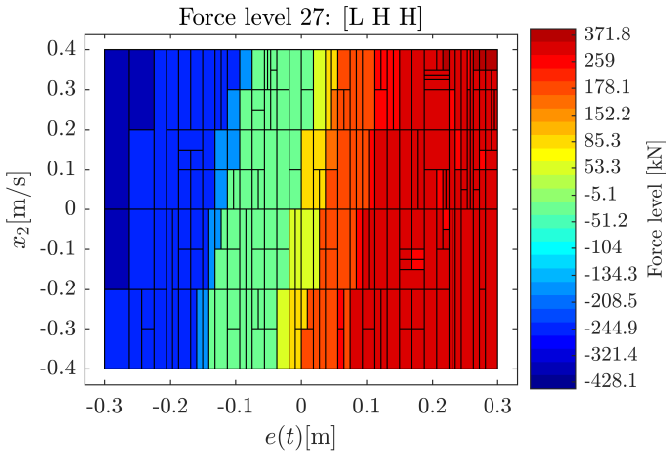



Force level 3: [H M L]
Force level 4: [H L H]
Force level 5: [H M M]
Force level 6: [M L L]
Force level 7: [H H L]
Force level 8: [H M H]
x_2[m/s]
$e(t)$[m]
Force level [kN]
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1


Force level 9: [M L M]
Force level 10: [H H M]
Force level 11: [M M L]
Force level 12: [M L H]
Force level 13: [H H H]
Force level 14: [M M M]
x_2[m/s]
$e(t)$[m]
Force level [kN]
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1

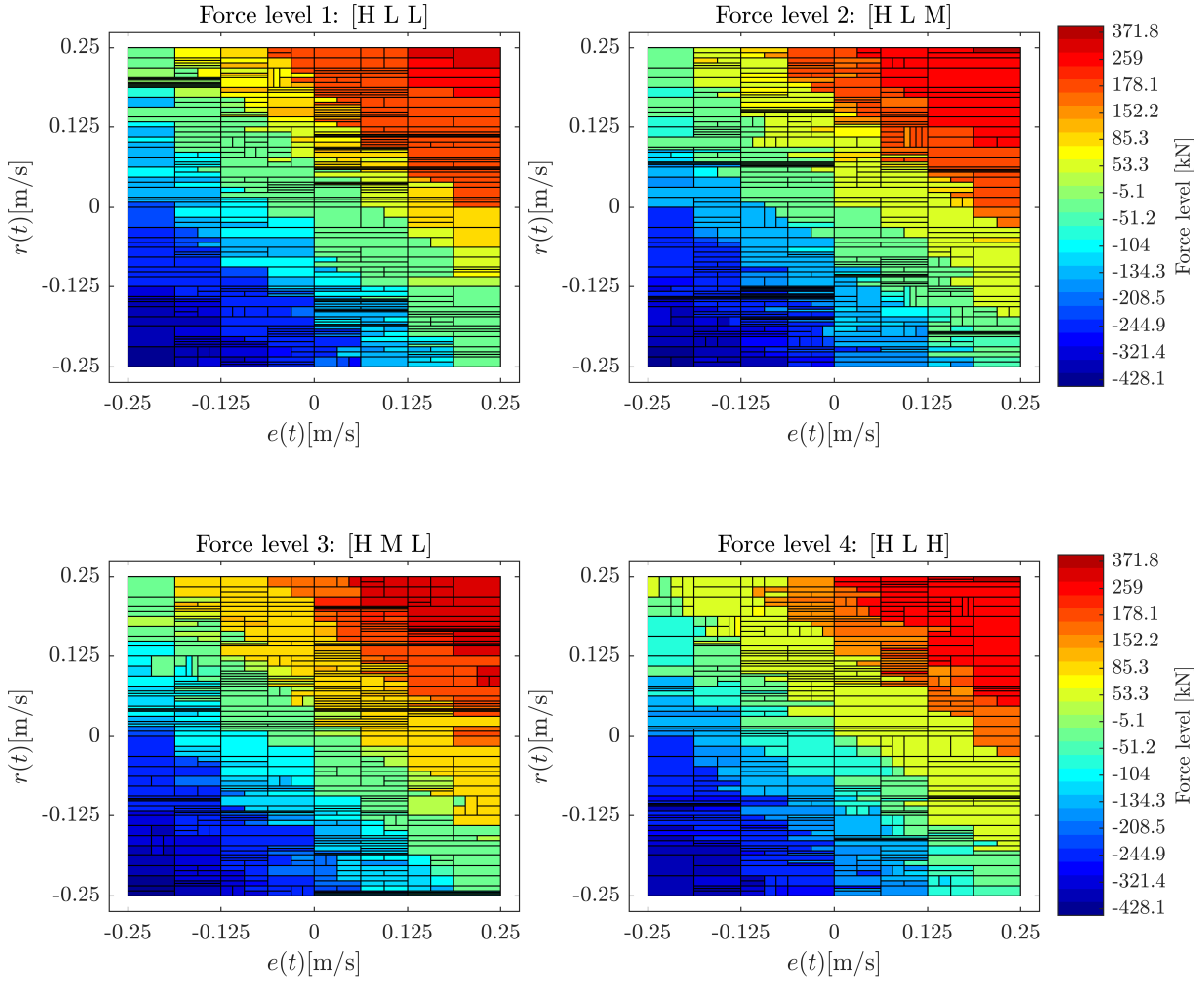

Force level 15: [L L L]
Force level 16: [M H L]
Force level 17: [M M H]
Force level 18: [L L M]
Force level 19: [M H M]
Force level 20: [L M L]
x_2[m/s]
$e(t)$[m]
Force level [kN]
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1

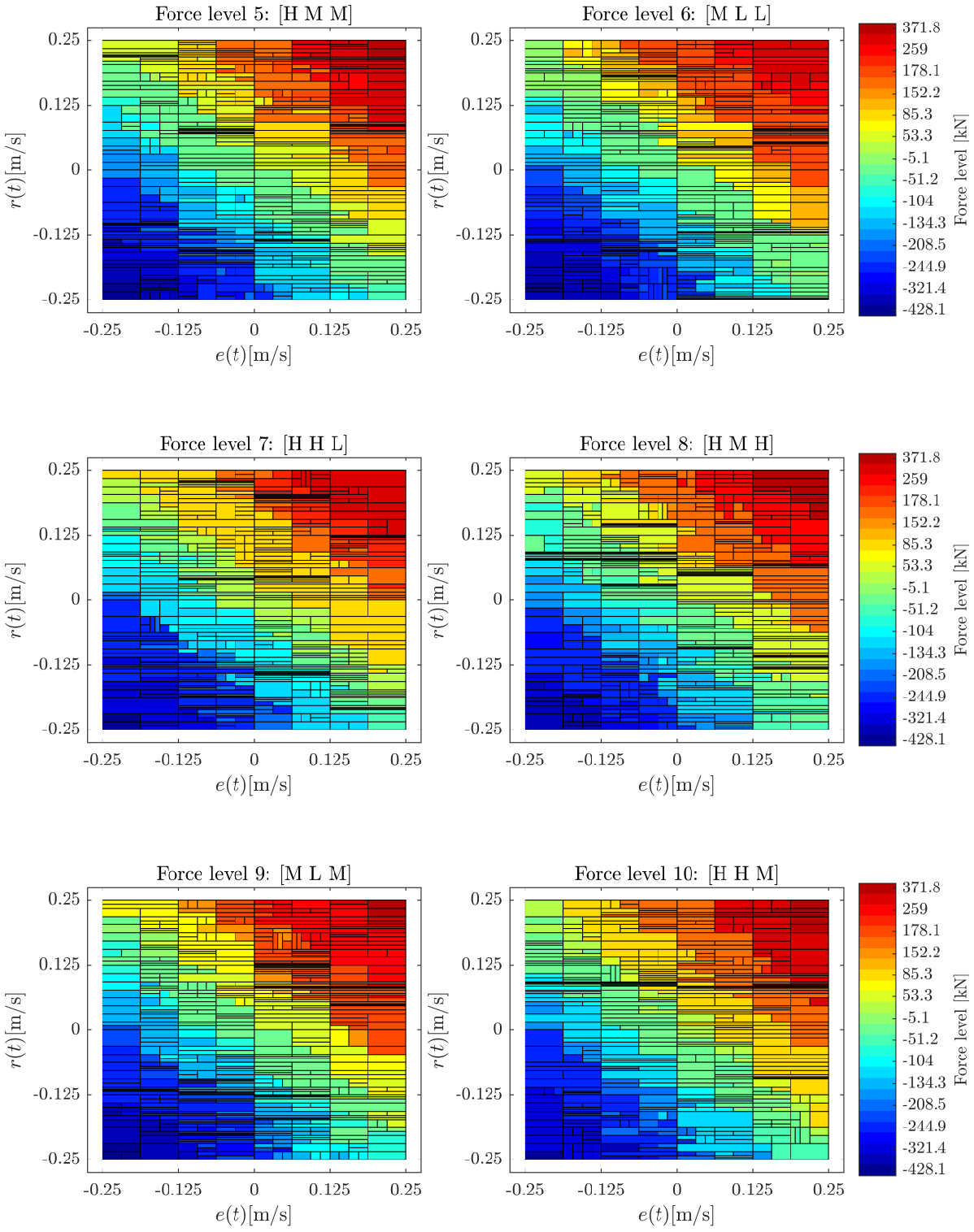

Force level 21: [L L H]
Force level 22: [M H H]
Force level 23: [L M M]
Force level 24: [L H L]
Force level 25: [L M H]
Force level 26: [L H M]
x_2[m/s]
$e(t)$[m]
Force level [kN]
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1

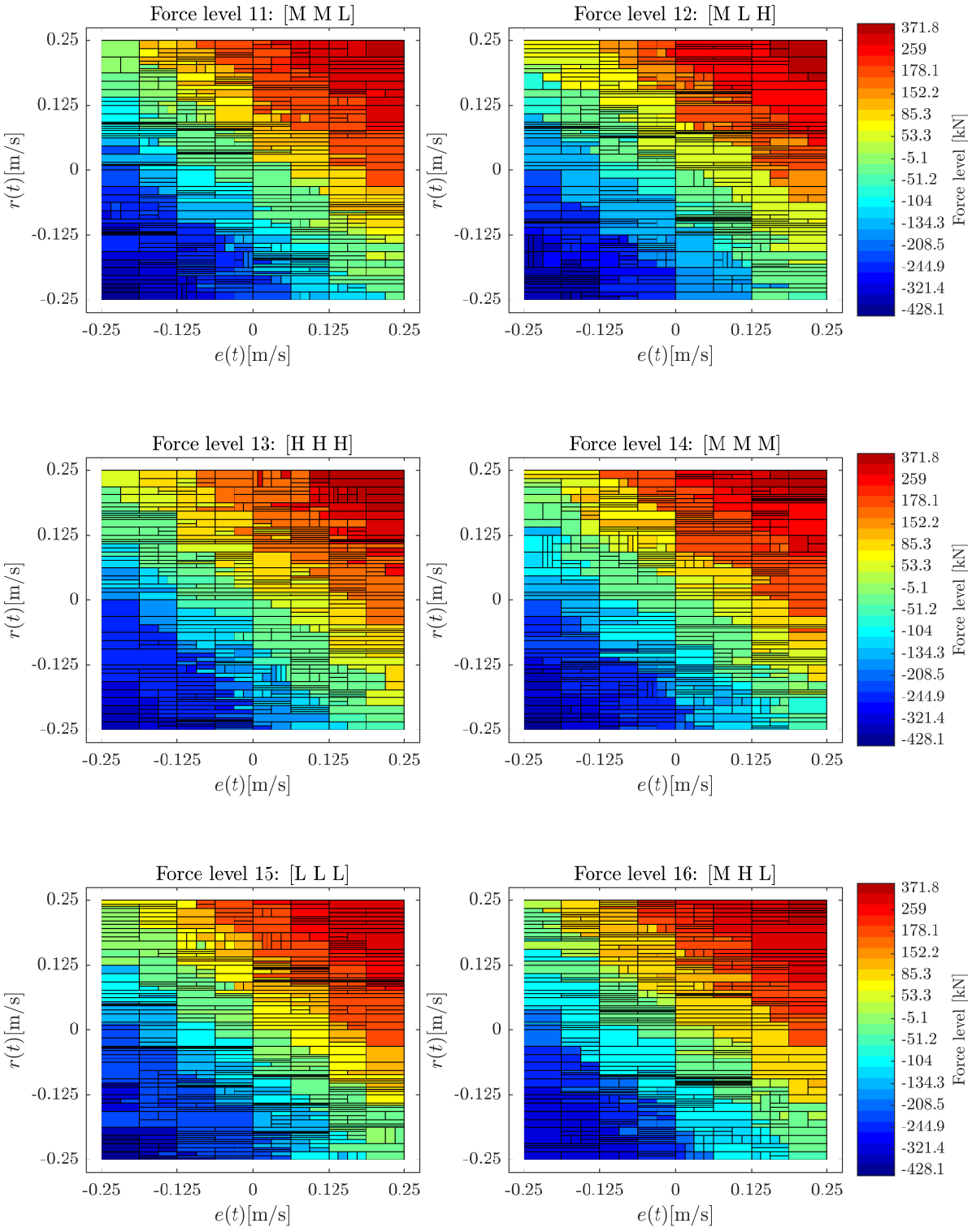


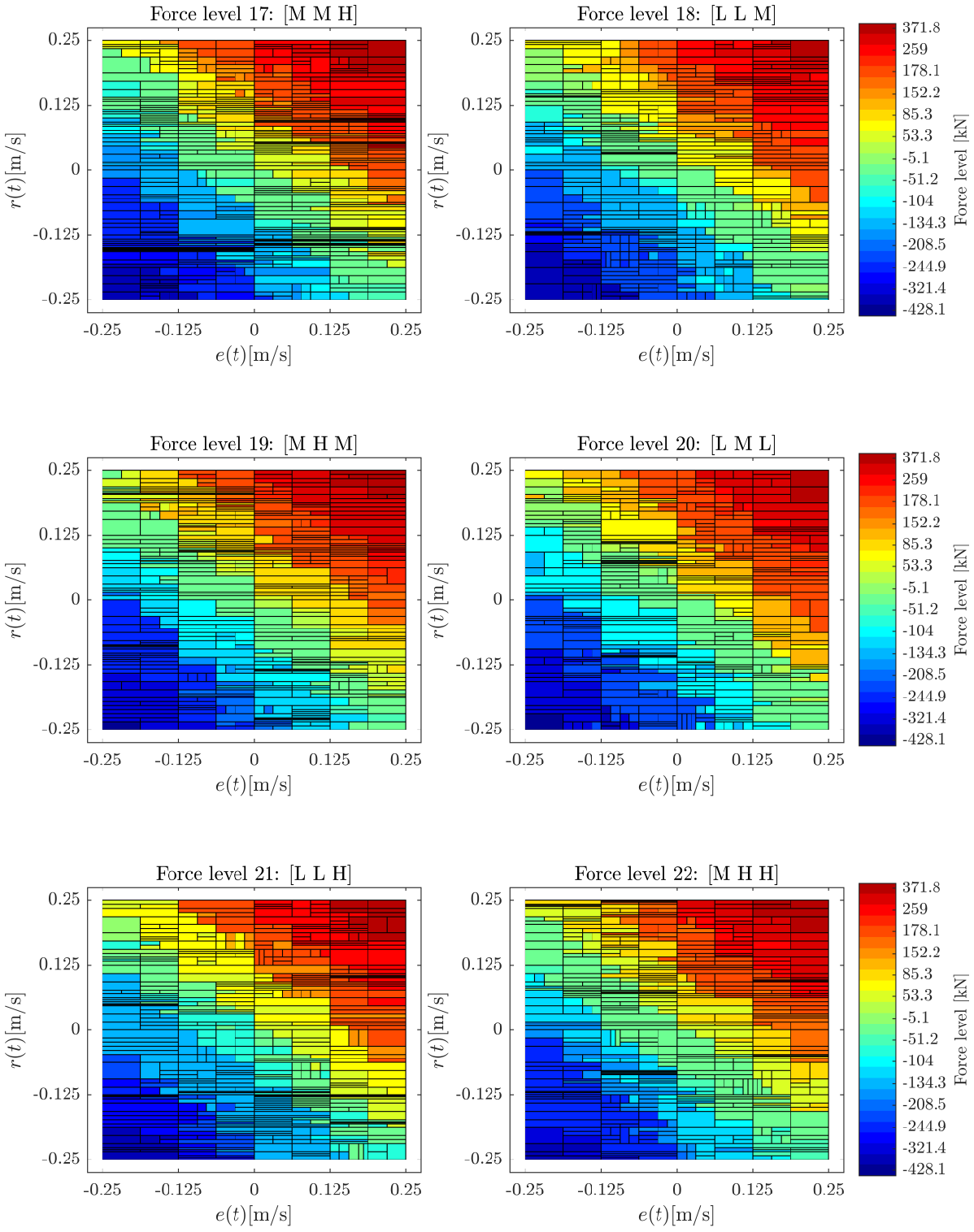
E.2 Solution for Explicit Model Predictive Velocity Control

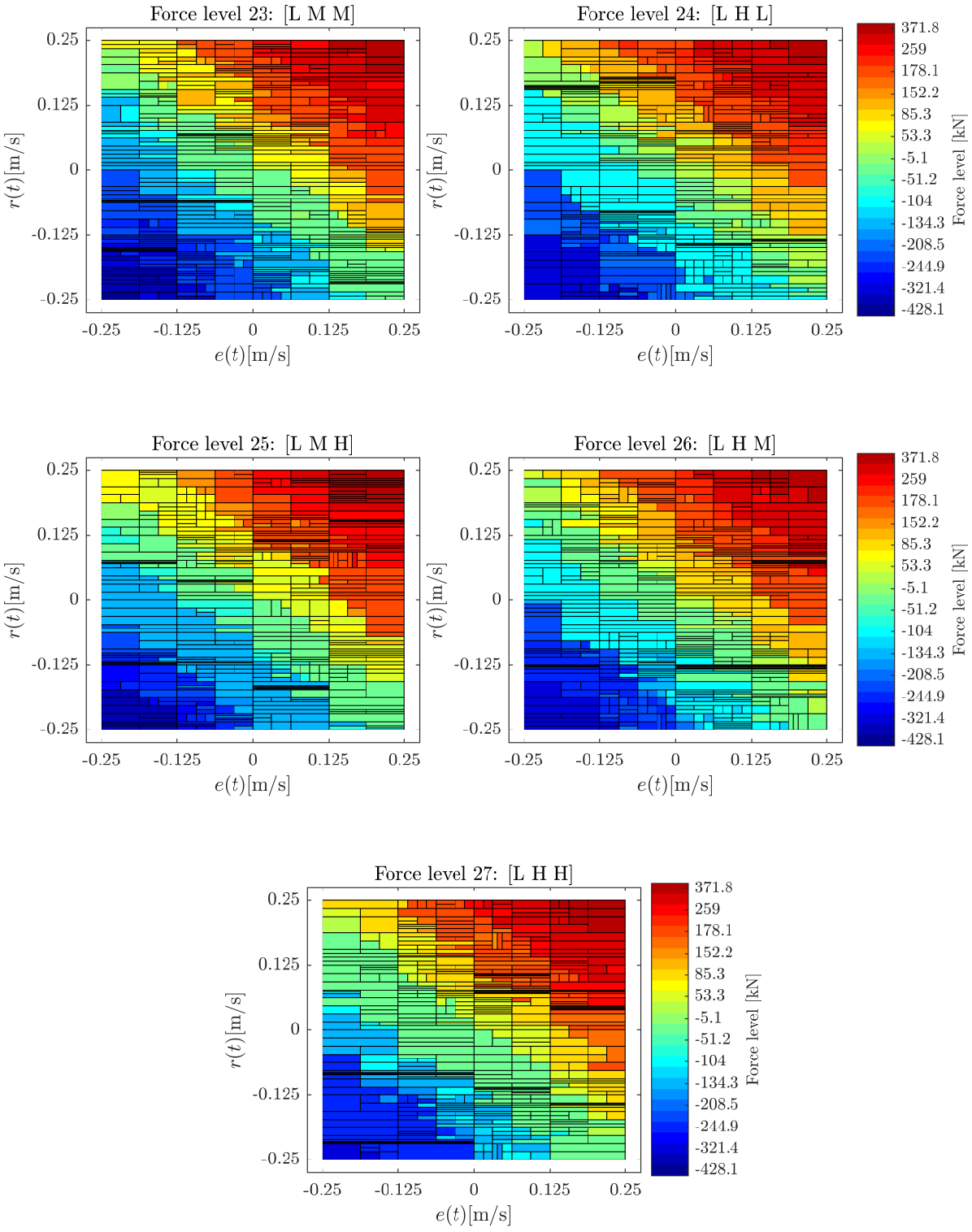
The explicite solution for the velocity control problem has the two continuous dimensions, which is the velocity error and the reference. The position is assumed constant at the center position. A solution is shown for each of the 27 force levels, which can be seen as the discrete dimension.




Force level 5: [H M M]
Force level 6: [M L L]
Force level 7: [H H L]
Force level 8: [H M H]
Force level 9: [M L M]
Force level 10: [H H M]
r(t)[m/s]
e(t)[m/s]
0.25
0.125
0
-0.125
-0.25
Force level [kN]
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1


Force level 11: [M M L]
Force level 12: [M L H]
Force level 13: [H H H]
Force level 14: [M M M]
Force level 15: [L L L]
Force level 16: [M H L]
$r(t)$[m/s]
$e(t)$[m/s]
0.25
0.125
0
-0.125
-0.25
Force level [kN]
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1


Force level 17: [M M H]
Force level 18: [L L M]
Force level 19: [M H M]
Force level 20: [L M L]
Force level 21: [L L H]
Force level 22: [M H H]
r(t)[m/s]
e(t)[m/s]
0.25
0.125
0
-0.125
-0.25
Force level [kN]
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1


Force level 23: [L M M]
Force level 24: [L H L]
Force level 25: [L M H]
Force level 26: [L H M]
Force level 27: [L H H]
r(t)[m/s]
e(t)[m/s]
0.25
0.125
0
-0.125
-0.25
371.8
259
178.1
152.2
85.3
53.3
-5.1
-51.2
-104
-134.3
-208.5
-244.9
-321.4
-428.1
Force level [kN]

APPENDIX F

LOAD SIDE LINEARISATION AND CONTROL

In this appendix, a pressure control for the load side cylinder will be synthesised based on linear control theory. Even though this is not directly in the scope of this thesis, pressure/force control of the load side is a fundamental step in order to use the test setup described in chapter 3 for further evaluation of motion control on the Discrete Displacement Cylinder.

As different test case application varies in load profile, different types of controllers are developed, mainly for tracking and regulating. A position tracking control was developed in [3] based on a similar approach. Before linearising the load cylinder equations, the following system simplifications are made:

- Supply and tank pressures are considered constant.
- Safety valves and accumulators are neglected.
- The two parallel valve configuration is considered as a single symmetric valve SISO system.

For simplicity the hydraulic model is derived only as a reduced order model where a load pressure is introduced. When describing the hydraulic model as a reduced order model the pressure dynamics in each chamber are not considered, but only the combined dynamic of the system. As the cylinder has a stroke length of 3m and is operated only ±0.5m from center position, this simplification is considered reasonable. The load flow and pressure are defined as

$$Q_L = \frac{1}{2}\left(Q_{cA} + Q_{cB}\right) \tag{F.1}$$

$$p_L = p_{cA} - p_{cB} \tag{F.2}$$

Assuming equal displacement flows in the cylinder ($Q_{cA} = Q_{cB}$), the load flow can be described as a function of the load pressure, yielding the following load flow orifice equation

$$Q_L = \begin{cases} c_d A(x_v)\sqrt{\frac{2}{\rho}}\sqrt{\frac{p_S - p_L - p_T}{2}} & \text{for} \quad x_v \geq 0 \\ c_d A(x_v)\sqrt{\frac{2}{\rho}}\sqrt{\frac{p_S - p_L + p_T}{2}} & \text{for} \quad x_v < 0 \end{cases} \tag{F.3}$$

Linearising this equation yields, for $x_v \geq 0$:

$$Q_L\left(x_v, p_L\right) = K_{qx}x_v - K_{qp}p_L \tag{F.4}$$

where

$$K_{qx} = \left.\frac{\partial Q_L}{\partial x_v}\right|_{x_{v0},p_{L0}} = c_d A_{d0}\sqrt{\frac{2}{\rho}}\sqrt{\frac{p_P - p_{L0} - p_T}{2}} \tag{F.5}$$

$$K_{qp} = \left.\frac{\partial Q_L}{\partial p_L}\right|_{x_{v0},p_{L0}} = \frac{c_d A_{d0}\sqrt{\frac{2}{\rho}}}{\sqrt{8(p_P - p_{L0} - p_T)}} \tag{F.6}$$

Subscript 0 indicates point of linearisation. Since the valves in the two-valve configuration have similar dynamic behavior, they are here treated as a single linear SISO system with second order dynamics:

$$G_{v,LS}(s) = \frac{x_v}{x_v^*} = \frac{\omega_{nv,LS}^2}{s^2 + 2\zeta_{v,LS}\omega_{nv,LS} + \omega_{nv,LS}^2} \tag{F.7}$$

where $\omega_{nv,LS} = 330$ rad/s and $\zeta_{v,LS} = 0.707$. A continuity equation can be derived based on the definition of load flow:

$$Q_L = \frac{Q_{cA} + Q_{cB}}{2} = -A_{cLS}\dot{x}_s + c_{leak}(p_{cA} - p_{cB}) + \frac{V_A(x_s)}{2\beta_e(p_{cA})}\dot{p}_{cA} + \frac{V_B(x_s)}{2\beta_e(p_{cB})}\dot{p}_{cB} \tag{F.8}$$

To simplify this expression, the point of linearisation is chosen to be at the cylinder center position. Thus, for this symmetrical cylinder, the two chamber volumes are equal. Furthermore it is assumed that $\beta_e(p_{cA}) = \beta_e(p_{cB}) = \beta_0 = cst.$, and that the position dependent term of $V_A(x_s)$ and $V_B(x_s)$ is negligible. This reduces the continuity equation to the linear equation

$$\dot{p}_L = \frac{4\beta_0}{V_{t,LS}}\left(Q_L + A_{cLS}\dot{x}_s - c_{leak}p_L\right) \tag{F.9}$$

where $V_{t,LS} = V_{cA} + V_{cB}$ is the total load cylinder volume. The choice of linearisation point for the cylinder position yields the lowest system eigenfrequency. Choosing a linearisation point for the valve position (x_{v0}) affects system damping term through the flow-pressure coefficient K_{qp}. A small valve opening yields lower system damping and thus leads to a conservative control design. The mechanical system is linearised by ignoring the non-linear Coulomb-friction of equation 5.27 on page 27, yielding the linear system

$$m_s\ddot{x}_s = F_{DDC} - F_{LS} - B_{vs}\dot{x}_s \tag{F.10}$$

$$= F_{DDC} - A_{cLS}p_L - B_{vs}\dot{x}_s \tag{F.11}$$

The above equations is put into state space form with state vector $\mathbf{x}_{LS} = \begin{bmatrix} x_s, & \dot{x}_s & p_L & x_v & \dot{x}_v \end{bmatrix}^T$ and input $u = x_v^*$:

$$\dot{\mathbf{x}}_{LS} = \mathbf{A}_{LS}\mathbf{x}_{LS} + \mathbf{B}_{LS}u_{LS} \quad , \quad y = \mathbf{C}\mathbf{x}_{LS} \tag{F.12}$$

where

$$\mathbf{A}_{LS} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{B_{vs}}{m_s} & -\frac{A_{cLS}}{m_s} & 0 & 0 \\ 0 & \Lambda A_{cLS} & -\Lambda(K_{qp} + c_{leak}) & \Lambda K_{qx} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_{nv,LS}^2 & -2\zeta_{v,LS}\omega_{nv,LS} \end{bmatrix}, \quad \mathbf{B}_{LS} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_{nv,LS}^2 \end{bmatrix} \tag{F.13}$$

$$\mathbf{C}_{LS} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{F.14}$$

where $\Lambda = \frac{4\beta_0}{V_{t,LS}}$. The open loop transfer function is then:

$$G_{LS}(s) = \frac{p_L}{x_v^*} = (sI - A_{LS})^{-1} B_{LS} \tag{F.15}$$

$$= \frac{\frac{K_{qx}}{m_s}\left(s + \frac{B_{vs}}{m_s}\right)}{s^2 + \left(\frac{B_v}{m_s} + \Lambda\left(c_{leak} + K_{qp}\right)\right)s + \frac{\Lambda}{m_s}\left(A_{cLS}^2 + B_{vs}\left(c_{leak} + K_{qp}\right)\right)} G_{v,LS}(s) \tag{F.16}$$

The damping ratio $\zeta_{h,LS}$ and eigenfrequency $\omega_{nh,LS}$ of the mechanical-hydraulic system is then found to be:

$$\omega_{nh,LS} = \sqrt{\frac{\Lambda}{m_s}\left(A_{cLS}^2 + B_{vs}(c_{leak} + K_{qp})\right)} \quad , \quad \zeta_{h,LS} = \frac{1}{2\omega_{nh,LS}}\left(\frac{B_{vs}}{m_s} + \Lambda(c_{leak} + K_{qp})\right) \tag{F.17}$$

The undamped eigenfrequency is found to be approximately 10 Hz. Due to the low mass of the system and large volume of the cylinder chambers, $\frac{B_{vs}}{m_s} \gg \Lambda\left(c_{leak} + K_{qp}\right)$, thus the system damping is mainly due to the mechanical feedback term. The damping ratio is found to be approximately 0.09, which indicates a highly underdamped system resulting in a resonance peak around the eigenfrequency. Furthermore, the mechanical feedback term contributes with a zero in the transfer function, causing the system to further amplify inputs at frequencies larger than $\frac{B_{vs}}{m_s} \approx 1.7$ Hz, thus adding to the amplification of the resonance peak. The bode plot of G_{LS} can be seen in in figure F.1.

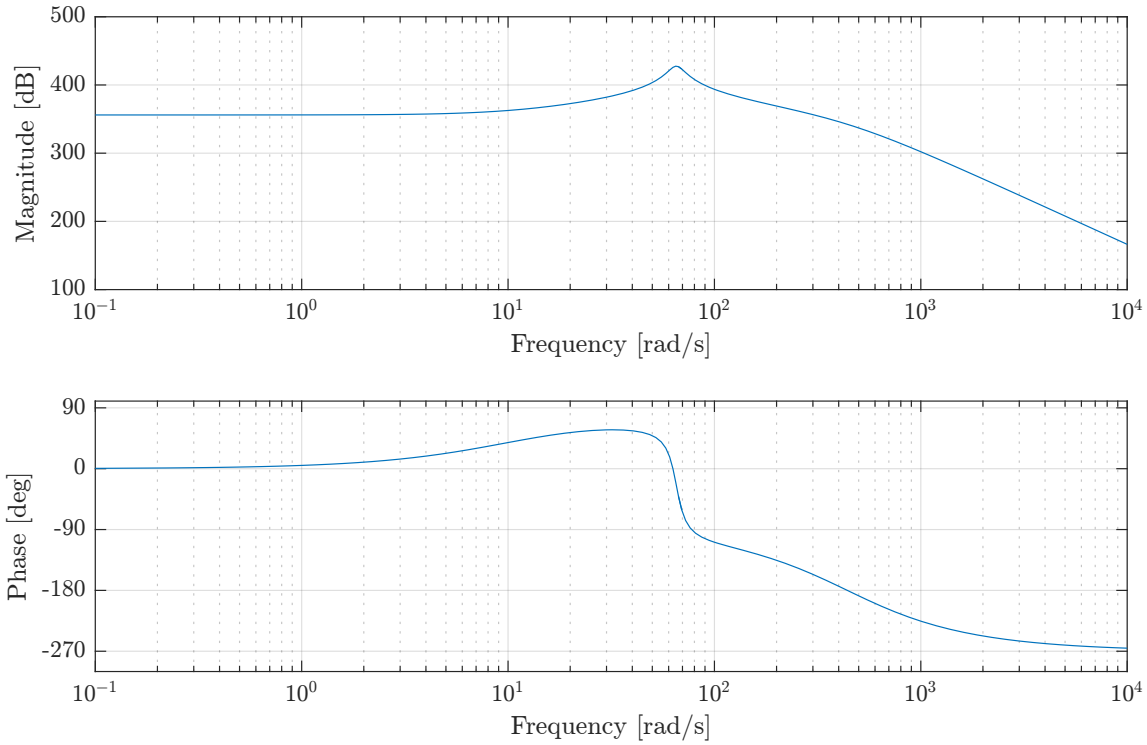


Figure F.1. Bode plot of G_{LS}.

F.1 Linear Model Validation

Validation of the linear model is done by comparing outputs with the non-linear model to similar input and working conditions. The linear model is validated at half stroke length and under the

following operation points:

$$x_{v0} = 0.032[-] \quad p_{L0} = 67[\text{bar}] \tag{F.18}$$

The results are shown in figures F.2 and F.3.

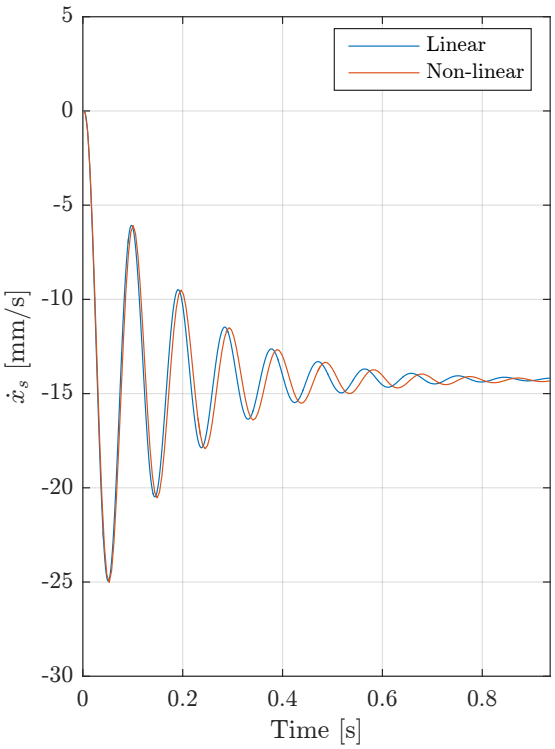


Figure F.2. Velocity validation.

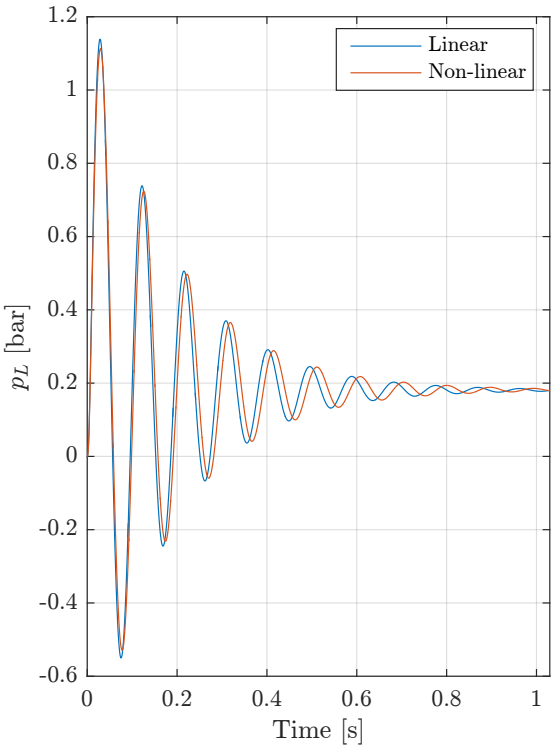


Figure F.3. Load pressure validation.

It can be seen that the eigenfrequency and steady state gain of the linear and non-linear model are similar. Hence, it is concluded that the linear model is sufficient for control design.

F.2 Valve Non-linearity Cancellation

As was seen from the linearisation, the orifice equation is the main source of non-linearities in the system. To cancel this non-linearity, the detailed valve model derived in chapter 5 is used to modify the system to calculate an equivalent spool position based on the required flow by inverting the flow gain:

$$Q_L^* = c_d A(x_v) \sqrt{\frac{2}{\rho}} \sqrt{\Delta p} \tag{F.19}$$

$$\Leftrightarrow x_v^* = \frac{Q_L^*}{c_d A_0 \sqrt{\frac{2}{\rho}} \sqrt{\Delta p}} \tag{F.20}$$

where Δp is $p_P - p_T - p_L$ or $p_P - p_T + p_L$ depending on active metering edge. To analyse the effects of the load pressure feedback the equation is linearised, yielding

$$Q_L^* = K_{qx} x_v - K_{qp} p_L \tag{F.21}$$

$$\Leftrightarrow x_v^* = \frac{Q_L^*}{K_{qx}} + \frac{K_{qp}}{K_{qx}} p_L \tag{F.22}$$

The modified state space equations are then

$$\mathbf{A}_{LS} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{B_{vs}}{m_s} & -\frac{A_{cLS}}{m_s} & 0 & 0 \\ 0 & \frac{4\beta_0}{V_{t,LS}} A_{cLS} & -\frac{4\beta_0}{V_{t,LS}}(K_{qp} + c_{leak}) & \frac{4\beta_0}{V_{t,LS}} K_{qx} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{K_{qp}}{K_{qx}} & -\omega_{nv,LS}^2 & -2\zeta_{v,LS}\omega_{nv,LS} \end{bmatrix}, \quad \mathbf{B}_{LS} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{\omega_{nv,LS}^2}{K_{qx}} \end{bmatrix} \tag{F.23}$$

From this, it can be seen that the load pressure feedback is positive and effectively decreasing system damping. To reduce this effect, a load pressure reference can instead be used, transforming the term into a feed forward gain and thus acting as a pre-filter on the load pressure reference. For load pressure control, assuming a small control error, this will only result in a small error in the feed forward gain.

F.3 Linear Quadratic Regulator

The control problem is solved using optimal control. Linear Quadratic Regulator is an optimisation approach based linear state feedback control. The criterion to be minimised is a weighting of control input energy and control error. Thus the cost function can be described as F.24 for the linear system in F.23.

$$J = \int_0^\infty \left(\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) \right) dt \tag{F.24}$$

The state vector used is $\mathbf{x} = \begin{bmatrix} \dot{x}_s & p_L & x_v & \dot{x}_v \end{bmatrix}^T$, as position has no influence on pressure dynamics in the linearised system. Here, J is the cost function and $\mathbf{Q}$ and R are the weighting for the state errors and inputs respectively. $\mathbf{Q}$ should be chosen positive semidefinite while $\mathbf{R}$ should be chosen semidefinite. With only one input, $\mathbf{R}$ is a scalar and is set to unity. The penalty is then tuned by adjusting the elements of $\mathbf{Q}$. The optimisation problem of F.24 is minimised by solving the stationary Ricatti equation in MATLAB, yielding a set of feedback gains $\mathbf{L}$, and the control law is thus

$$u(t) = -\mathbf{L}\mathbf{x}(t) \tag{F.25}$$

For a tracking problem where the controlled variable should follow a given reference signal the control law becomes

$$u(t) = -\mathbf{L}\mathbf{x}(t) + L_r r(t) \tag{F.26}$$

$$= -L_1 x_1(t) - L_2 x_2(t) - L_3 x_3(t) - L_4 x_4(t) + L_r r(t) \tag{F.27}$$

where the feed forward gain L_r is chosen to be equal to L_2 to give unity steady state gain for pressure control. The spool position is not readily available as a measurement. Instead, the spool position

is estimated from the desired load flow Q_L^* using the valve dynamics and inverse orifice equation described above. The effect of this spool observer can be analysed by linearisation of the observer. The output of the control is as equation F.27. Ignoring load pressure and velocity feedback, the control output is then

$$u = -L_3 x_3 - L_4 x_4 + L_r r \tag{F.28}$$

where

$$x_3 = \frac{\omega_{nv,LS}^2}{s^2 + 2\zeta_{v,LS}\omega_{nv,LS} + \omega_{nv,LS}^2} \frac{u}{K_{qx}} \tag{F.29}$$

$$x_4 = s x_3 \tag{F.30}$$

Inserting this yields

$$u = -\frac{\omega_{nv,LS}^2 (L_3 + sL_4}{s^2 + 2\zeta_{v,LS}\omega_{nv,LS} + \omega_{nv,LS}^2} \frac{u}{K_{qx}} + L_r r \tag{F.31}$$

$$\Leftrightarrow \frac{u}{r} = K_{qx} L_r \frac{s^2 + 2\zeta_{v,LS}\omega_{nv,LS} + \omega_{nv,LS}^2}{s^2 + (2\zeta_{v,LS}\omega_{nv,LS} + \omega_{nv,LS}^2 L_4)s + \omega_{nv,LS}^2 (1 + L_3)} \tag{F.32}$$

Thus, for L_3, $L_4 > 0$ the observer acts as a second order lead regulator, which allows attenuation of high frequency dynamics in the system. Initially, the LQR-problem is solved with a penalty solely on the load pressure. This is tuned to achieve the desired bandwidth. This reveal; $\mathbf{Q} = \text{diag}(0, 10^{-8}, 0, 0)$. Furthermore, this results in a more dampened system, with a damping factor of 0.5. For further tuning $\mathbf{Q}$, the effect of increasing feedback gains on the closed loop system, root locus for all four states are plotted in figures F.4-F.7. This gives an idea of which states to penalise to achieve wanted characteristics. As seen in F.6 increasing feedback gain on spool position may increase damping until a certain point. Rate feedback in form of spool position is seen to dampen the system, but also moving the less dominant pole closer to the imaginary axis.

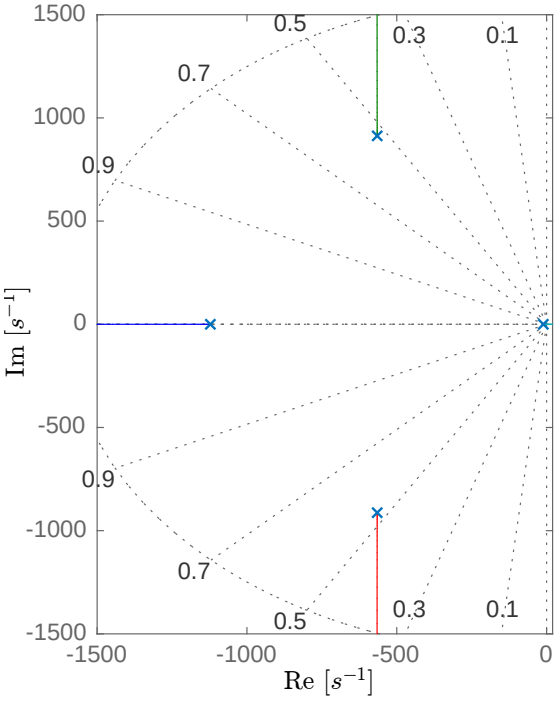


Figure F.4. Root locus for velocity feedback

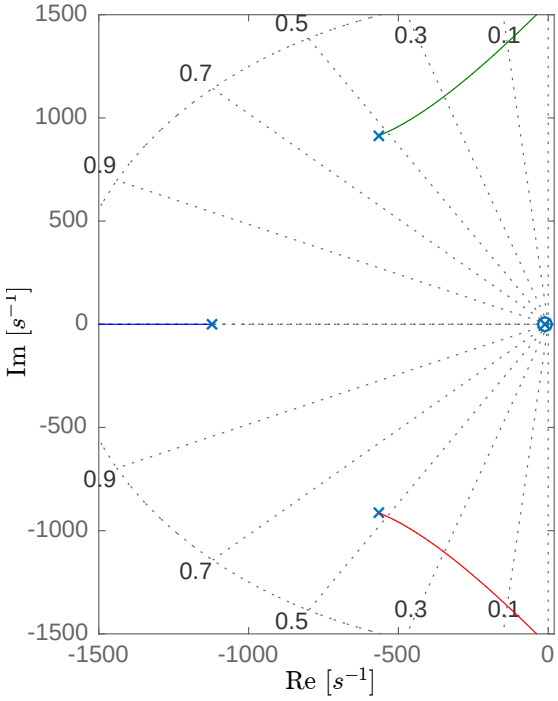


Figure F.5. Root locus for pressure feedback

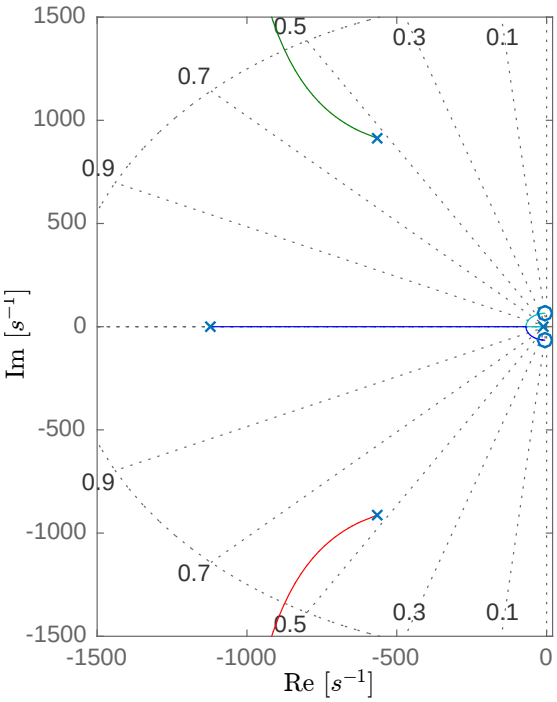


Figure F.6. Root locus for spool position feedback

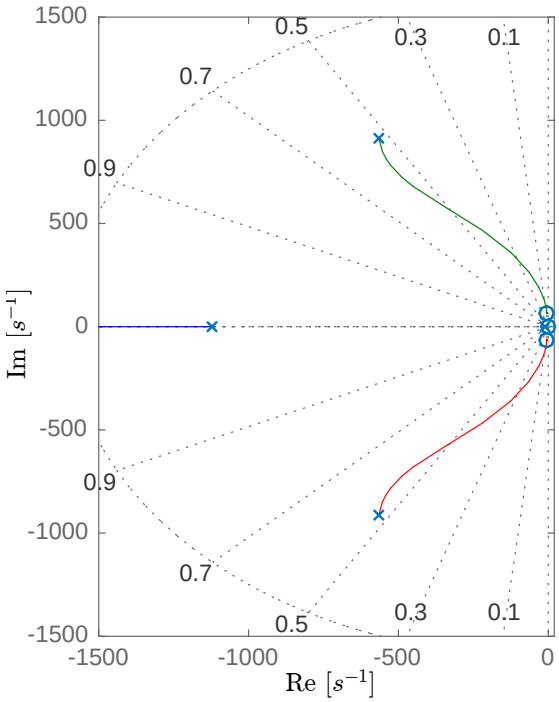


Figure F.7. Root locus for spool velocity feedback

The load pressure is the state to controlled. Hence, it is not the load force exerted on the DDC that is being controlled as no friction compensation has been developed. However, due to the large forces exerted, the friction contribution is relatively small and is considered negligible.

F.4 Velocity feed forward

The piston velocity can be interpreted as a disturbance on the load pressure for which the control need to compensate. Since, for position or velocity control of the DDC, a velocity trajectory is often available. This velocity trajectory can be mapped to a required displacement flow, cancelling the velocity disturbance (assuming a small velocity error). The required displacement flow feed forward is then

$$Q_{\dot{x}_s,ff} = \dot{x}_s^* A_{cLS} \tag{F.33}$$

The flow command $Q_{\dot{x}_s,ff}$ is then added to the control output of the LQR-control.

F.5 LQR Integrator

The steady state gain of the closed loop system using the above control is unity, causing the system to track the reference. However, model errors and disturbances are not well compensated and thus a higher steady state gain is desirable. For a constant or slowly varying reference, the state space model can be augmented to include an integrator state $\dot{z} = p_L$. The state vector is then $\tilde{\mathbf{x}}_{LS} = \begin{bmatrix} x_s, & \dot{x}_s, & p_L, & x_v, & \dot{x}_v, & z \end{bmatrix}^T$ The augmented state space equations are then

$$\tilde{\mathbf{A}}_{LS} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \quad , \quad \tilde{\mathbf{B}}_{LS} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \tag{F.34}$$

where $\mathbf{0}$ here are zero-vectors of appropriate size. The extra state can then be penalised in a similar manner to control the integrator gain. The required bandwidth is achieved with $\tilde{\mathbf{Q}} = \text{diag}(0, 10^{-8}, 0, 0, 10^6)$.

APPENDIX G

SIGNAL SPLITTING BETWEEN THE VALVES

The signal to each of the two proportional valves on the load side is the calculated using a "splitting algorithm". In this appendix two splitting algorithms are explained. First, the one already implemented on the test-setup, based on splitting the opening area. After this a revisited splitting algorithm with a more simple and smooth transaction on the threshold between the two valves. This second one is based on splitting an equivalent spool position.

G.1 Splitting Opening Area - Algorithm

The first splitting algorithm is the one implemented by [3].

One of the issue with this algorithm is that the ramp on spool position used to compensate for the dead band introduces a step in spool velocity. This might cause some problems later on if using an observer to feedback valve states. Another issue with this approach is the uncertainties about the opening characteristics given by the datasheet. There might be a further issue with the codependency of the two signals. All these issues will potentially result in some unpredictable oscillations around the dead band threshold.

Since the output of the controller is a flow reference the first step of the algorithm is to calculate a wanted opening area, $A_{o,ref}$, using the inverse orifice equation G.1.

$$A_{o,ref} = \frac{Q_{ref}}{c_d \sqrt{\frac{2}{\rho}\Delta p}} \tag{G.1}$$

Where ΔP is the pressure difference over the valve. The next step of the algorithm is a splitting of the opening area as in equation G.2. In figure G.1 the area reference to each valve is depicted.

$$\begin{aligned} A_{M,ref} &= \begin{cases} A_{o,ref} & , \quad A_{o,ref} \leq A_{o,threshold} \\ A_{o,threshold}(2 - \frac{A_{o,threshold}}{A_{o,ref}}) & , \quad A_{o,ref} > A_{o,threshold} \end{cases} \\ A_{P,ref} &= \begin{cases} 0 & , \quad A_{o,ref} \leq A_{o,threshold} \\ A_{o,ref} - A_{M,ref} & , \quad A_{o,ref} > A_{o,threshold} \end{cases} \end{aligned} \tag{G.2}$$

Where $A_{M,ref}$ and $A_{P,ref}$ is the wanted opening area of the MOOG and Parker valve respectively. $A_{o,threshold}$ is the opening threshold value for when the Parker-valve starts moving. For this case $A_{o,threshold} = A_{M,max} \cdot 0.25$. Where $A_{M,max}$ is maximum opening area of MOOG-valve.

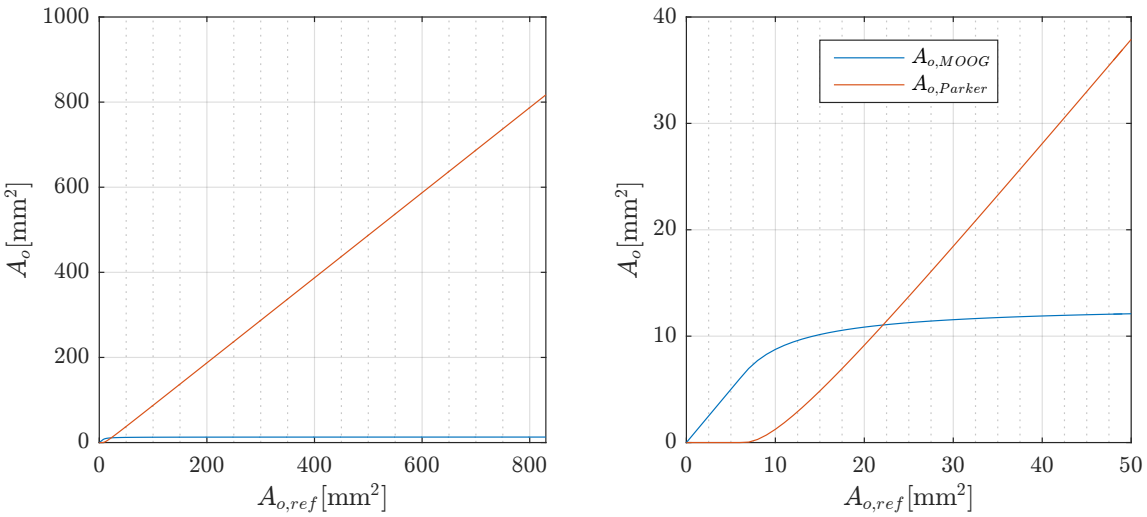


Figure G.1. Opening area reference to each valve as function of total wanted opening area.

The voltage signal to both valves should be in a range of [-10 : 10] v. For the MOOG-valve the voltage signal is found by G.3.

$$u_{moog} = 10\frac{A_{M,ref}}{A_{M,max}}\text{sgn}(Q_{ref}) \tag{G.3}$$

The voltage signal for the Parker valve is calculated by using a 3rd order polynomial to compensate for the non-linearity of the opening area as seen. There is also implemented an extra offset to the polynomial to account for the dead band. This is done with a ramp since the exact value of the dead band is unknown. In figure G.2 the voltage signal for a positive flow reference as function of the area reference given to each valve is shown. The voltage signal is multiplied with the sign of the flow reference.

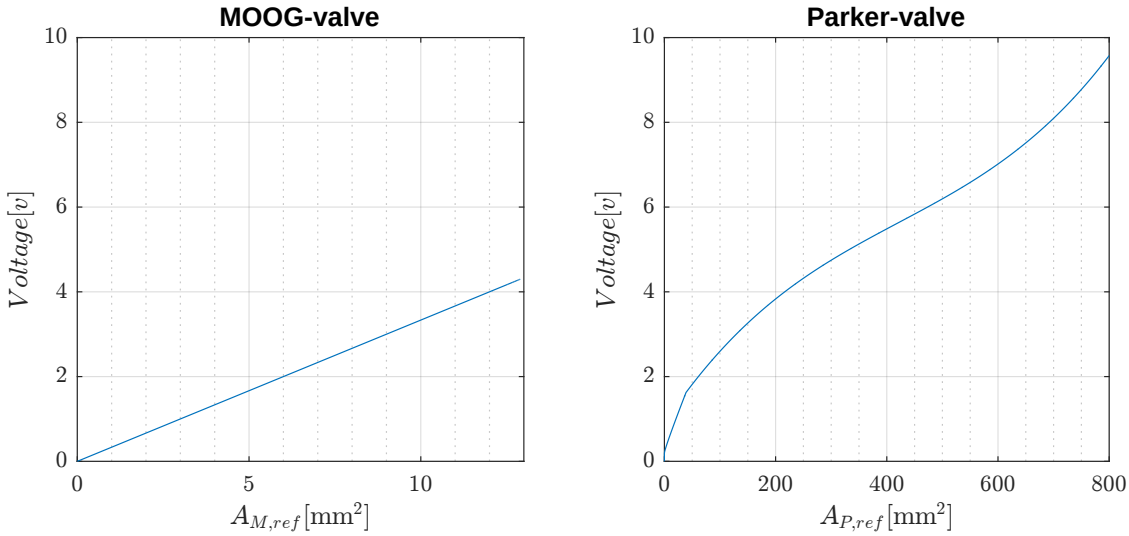


Figure G.2. Voltage signals as function of area reference to each valve.

G.2 Splitting Equivalent Spool Position - Algorithm

For this splitting algorithm the idea is to make a simple splitting algorithm to avoid the issues described for the "Area split" above. The disadvantage of this method is that it will not account for the non-linearities of the Parker-valve. These will then have to be compensated by the system controller.

Like the first algorithm an inverse orifice equation is used to find an equivalent spool position based on the flow reference, Q_{ref}.

$$x_{v,eq} = \frac{Q_{ref}}{A_{o,max} c_d \sqrt{\frac{2}{\rho}\Delta p}} \tag{G.4}$$

Where $A_{o,max}$ is a sum of maximum opening areas for both valves.

The equivalent spool position $x_{v,eq}$ is split out into a spool reference for the MOOG and parker valve, $x_{v,P}$ and $x_{v,M}$ respectively, by using the algorithm in equation G.5.

$$\begin{aligned} x_{v,M} &= \begin{cases} x_{v,eq}\frac{c_{m,max}}{2c_{xv,lim}} & , \quad x_{v,eq} < c_{xv,lim} \\ \frac{c_{m,max}}{2}(2 - \frac{c_{xv,lim}}{|x_{v,eq}|})\text{sgn}(x_{v,eq}) & , \quad x_{v,eq} \geq c_{xv,lim} \end{cases} \\ x_{v,P} &= \begin{cases} x_{v,eq} & , \quad x_{v,eq} < c_{xv,lim} \\ x_{v,eq} & , \quad x_{v,eq} \geq c_{xv,lim} \end{cases} \end{aligned} \tag{G.5}$$

Where the constant $c_{m,max}$ determines how much of the MOOG valves capacity is utilised with a value between 0 and 1. $c_{xv,lim}$ is a constant determining the value of the deadband limit of, $x_{v,eq}$, for which the Parker valve gets an input. The equivalent opening area from the algorithm is depicted in figure G.3.

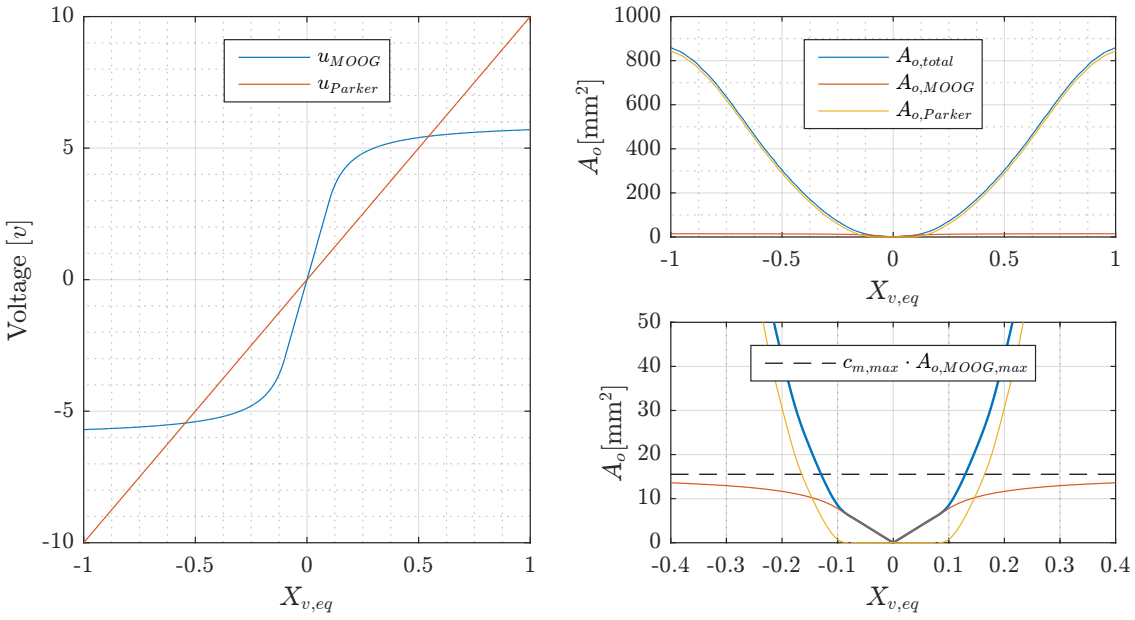


Figure G.3. To the left voltage signals as function of equivalent spool position. To the right opening area (according to data sheet information) as function of equivalent spool position.

G.3 Comparison of splitting algorithms

As mentioned previously both algorithms have pros and cons. Where the main disadvantage of the "area - split" is some discontinuity in spool velocity around the threshold value.

For simplicity the flow trough the valves can now be expressed with one orifice equation based on the equivalent valve reference, x_v.

$$Q_A = \begin{cases} k_{v,eq}(x_v)\sqrt{|p_P - p_A|}\text{sgn}(p_P - p_A) & , \quad x_v \geq 0 \\ -k_{v,eq}(x_v)\sqrt{|p_A - p_T|}\text{sgn}(p_A - p_T) & , \quad x_v < 0 \end{cases} \tag{G.6}$$

$$Q_B = \begin{cases} -k_{v,eq}(x_v)\sqrt{|p_B - p_T|}\text{sgn}(p_B - p_T) & , \quad x_v \geq 0 \\ k_{v,eq}(x_v)\sqrt{|p_P - p_B|}\text{sgn}(p_P - p_B) & , \quad x_v < 0 \end{cases} \tag{G.7}$$

Where $k_{v,eq}(x_v)$ is an equivalent valve gain as a function of valve reference. The valve reference x_v is either area reference or equivalent spool position. In figure G.4 the equivalent valve gain is illustrated. On the figure it is seen that the "area-split" algorithm theoretically is able to cancel the non-linearities of the Parker valve. But this is based on a lot of assumptions, one being that the algorithm is able to accurately predict which voltage corresponds to the wanted opening area.

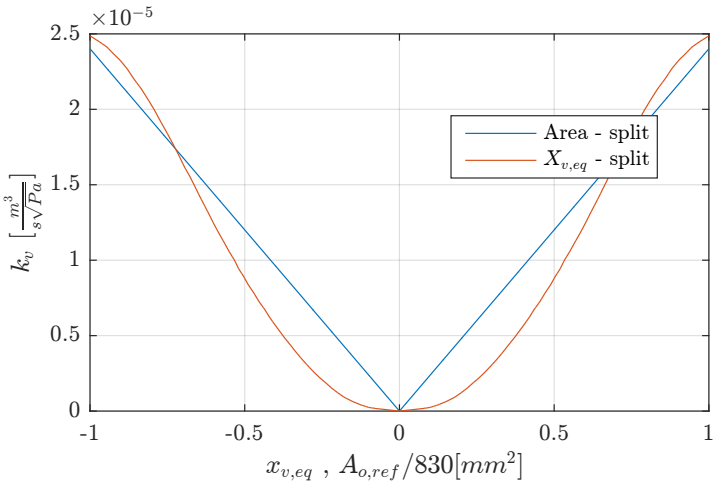


Figure G.4. Equivalent valve gain