

Evaluation of Control Algorithms for Hydraulic Pressure Control

Applied to Directional Control Valves in Mobile Machines

Master's thesis in Systems, Control and Mechatronics

DANIEL BÄCK
LARS-OLOF TURESSON

MASTER'S THESIS 2020

Evaluation of Control Algorithms for Hydraulic Pressure Control

Applied to Directional Control Valves in Mobile Machines

DANIEL BÄCK
LARS-OLOF TURESSON



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Evaluation of Control Algorithms for Hydraulic Pressure Control
Applied to Directional Control Valves in Mobile Machines
DANIEL BÄCK
LARS-OLOF TURESSON

© DANIEL BÄCK, 2020.

© LARS-OLOF TURESSON, 2020.

Supervisor: Marcus Mårlind, Parker Hannifin AB

Examiner: Torsten Wik, Department of Electrical Engineering

Master's Thesis 2020
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Cross section view of the K170LS directional control valve used in this thesis together with a simplified block diagram of the control system used to control the output pressure.

Printed by Chalmers Reproservice Gothenburg, Sweden 2020

Evaluation of Control Algorithms for Hydraulic Pressure Control
Applied to Directional Control Valves in Mobile Machines
DANIEL BÄCK
LARS-OLOF TURESSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Today the field of mobile hydraulics is moving more and more towards implementation of different software functions, but there are still many pure hydromechanical components used in the market. These components are often nonflexible and therefore manufacturers of mobile machines today request more advanced software solutions. One example of a pure hydromechanical component that may be replaced with a software function is the pressure feed reducer in a directional control valve. This component has the functionality to limit the maximum output pressure of the valve.

The objective of this master's thesis is to evaluate different control algorithms used for pressure control when replacing the mechanical pressure feed reducer with a software solution. A simplified model of the directional control valve is developed where the load is simplified as a constant volume. The model is used to design linear quadratic regulators with and without integral action together with \mathcal{H}_∞ controllers, a model predictive controller and a PID controller with bumpless transfer. Results from simulations of the controllers on the developed model are presented together with results from tests on the boom function of a backhoe.

The conclusion from the thesis is that the two controllers performing best are the linear quadratic regulator with integral action and the PID controller with bumpless transfer. Both reached the maximum pressure within a margin of 4 bar and had a satisfactory pressure behaviour. However, these controllers do not perform as well as the mechanical pressure feed reducer since they start to limit the pressure approximately 50 to 70 bar earlier in the worst case. This is due to the dynamics of the directional control valve being too slow to make the controllers perform as well as the mechanical feed reducer. Even if the controllers are not as fast as the mechanical pressure feed reducer, a desirable behaviour is obtained for the backhoe. In order to assess if this behaviour is good enough to apply on different mobile machines in practice, further tests and analysis are required.

Keywords: Pressure control, directional control valve, pressure feed reducer, LQR, \mathcal{H}_∞ , \mathcal{H}_∞ Loop-Shaping, MPC, PID.

Evaluering av regleralgoritmer för hydraulisk tryckreglering
Applicerat på riktningssystem i mobila maskiner
DANIEL BÄCK
LARS-OLOF TURESSON
Institutionen för Elektroteknik
Chalmers Tekniska Högskola

Sammanfattning

Branschen inom hydraulik för mobila maskiner går idag mer och mer mot implementering av olika mjukvarulösningar, men det finns fortfarande många rent hydromekaniska komponenter på marknaden. Dessa komponenter är oftast ej flexibla och därför efterfrågar tillverkare av mobila maskiner idag mer avancerade mjukvarulösningar. Ett exempel på en ren hydromekanisk komponent som skulle kunna ersättas av en mjukvarufunktion är matarreduceraren i en riktningssystem. Denna komponent har funktionen att begränsa det maximala utgående trycket från ventilen.

Målet med detta examensarbete är att designa och utvärdera olika regleralgoritmer som kan användas till tryckstyrning i en mjukvarufunktion för att ersätta den mekaniska matarreduceraren. En förenklad modell av en riktningssystem utvecklas där den anslutna lasten är förenklad till en konstant volym. Modellen används för att designa LQR-regulatorer med och utan integralverkan tillsammans med \mathcal{H}_∞ regulatorer, en modellprediktiv regulator och en PID regulator med mjuk övergång. Resultat från simuleringar av de olika regulatorerna på den utvecklade modellen presenteras tillsammans med resultat från tester på bomfunktionen till en grävlastare.

Slutsatsen är att de två regulatorerna som fungerar bäst är LQR-regulatorn med integralverkan och PID-regulatorn med mjuk övergång. Båda dessa nådde det maximala trycket inom en marginal på 4 bar och hade ett önskvärt tryckbeteende. Dock presterar dessa regulatorerna ej lika bra som den mekaniska matarreduceraren eftersom de börjar begränsa trycket cirka 50 till 70 bar tidigare i det värsta fallet. Detta på grund av att dynamiken i riktningssystemen är för långsam för att få regulatorerna att prestera lika bra som den mekaniska matarreduceraren. Även om regulatorerna inte är lika snabba som den mekaniska matarreduceraren så erhålls ett önskvärt beteende för grävlastaren. För att kunna avgöra om detta beteende är tillräckligt bra för att appliceras på olika mobila maskiner i praktiken krävs ytterligare tester och analyser.

Nyckelord: Tryckreglering, riktningssystem, matarreducerare, LQR, \mathcal{H}_∞ , \mathcal{H}_∞ Loop-Shaping, MPC, PID.

Acknowledgements

We would like to thank Parker Hannifin in Borås for letting us carry out this master's thesis and providing us with the necessary equipment. A special thanks goes to our supervisor Marcus Mårland for his big interest in the project and the constant support during these months. It has been a pleasure having you as supervisor, thanks!

Besides this, we would like to thank everybody at the Parker divisions in Borås and Mölnlycke that in some way have shown interest in our work or helped us with our master's thesis during this spring.

Finally, we want to thank our respective families for always showing support and encouraging us during both this master's thesis work and our studies.

Daniel Bäck & Lars-Olof Turesson, Gothenburg, May 2020

Contents

List of Figures	xiii
List of Tables	xvii
Nomenclature	xix
1 Introduction	1
1.1 Aim	2
1.2 Problem Formulation	2
1.3 Research Questions	3
1.4 Problem Discussion	3
1.5 Scope & Boundaries	4
1.6 Previous Works	4
1.7 System Overview	5
2 Directional Control Valve	7
2.1 Basic Functionality	7
2.2 Pressure Compensator	9
2.3 Mechanical Pressure Feed Reducer	10
2.4 The K170LS Valve Used for Implementation	11
3 Mathematical Modelling	13
3.1 Valve Model	14
3.1.1 Pilot Valve	14
3.1.2 Spool	14
3.1.3 Pressure Compensator	15
3.1.4 Hydraulic Flow	16
3.1.5 Verification of Spool Model	17
3.2 Load Model	18
3.3 State Space Representation	20
3.4 Linear Model for Control	20
4 Control Algorithms	23
4.1 Discretisation	23
4.2 Linear Quadratic Regulator	24
4.2.1 LQR for Pressure Feed Reducer	25
4.3 LQR with Integral Action	26

4.3.1	LQRI for Pressure Feed Reducer	28
4.4	\mathcal{H}_∞ Controller	30
4.4.1	\mathcal{H}_∞ Controller for Pressure Feed Reducer	32
4.5	\mathcal{H}_∞ Loop-Shaping	34
4.5.1	\mathcal{H}_∞ Loop Shaping for Pressure Feed Reducer	36
4.6	Model Predictive Controller	38
4.6.1	MPC for Pressure Feed Reducer	39
4.7	PID Controller with Bumpless Transfer	41
4.7.1	PID Controller for Pressure Feed Reducer	42
4.8	Controller Output Conversion	43
5	Simulation of Controllers	45
5.1	Model Used for Simulation	45
5.2	Simulation Results	46
6	Implementation	51
6.1	Controller Implementation	51
6.2	Measurement Filtering	51
6.3	Input Current Filtering	52
6.4	Variable Integral Gain	54
7	Results	55
7.1	Test Description	55
7.2	System Limitations	57
7.3	Controller Comparison	59
7.4	System Phenomena	66
7.4.1	Pressure Oscillations	66
7.4.2	Dropping Pressure	67
7.4.3	Varying Dynamics	68
8	Discussion	71
8.1	Controller Comparison	71
8.2	Varying Dynamics	74
8.3	Oscillations & Notch Filter	75
8.4	Dropping Pressure Phenomenon	75
8.5	Model	76
9	Conclusion	79
10	Future Work	81
	Bibliography	83

List of Figures

1.1	Schematic overview of the system used for implementation and tests. 1. Backhoe, 2. Boom cylinder, 3. Pressure sensors, 4. Joystick, 5. Touch display, 6. Controller module, 7. Directional control valve. From [7]–[12]. Adapted with permission.	5
2.1	Hydraulic scheme for a directional control valve with a closed-centre spool that is electrohydraulic pilot actuated and connected to a pump, tank and double acting cylinder. A and B are the two workports, P is the hydraulic pump and T is the tank.	8
2.2	An example of the spool opening area as a function of the spool position in a directional control valve. Note the deadband and that the function is non-linear.	9
2.3	A simplified hydraulic scheme of a directional control valve with a pressure compensator.	10
2.4	Cross section view of a K170LS directional control valve where the different components are labelled. From [12]. Adapted with permission.	11
3.1	Overview of the complete system to model and its different parts. . .	13
3.2	Schematic model of the spool in the directional control valve used to describe its dynamics.	15
3.3	Schematic model of the pressure compensator in the directional control valve used to describe its dynamics.	15
3.4	Comparison of valve model and black-box model for two different reference steps in pilot pressure. As can be seen, the valve model follows the verified black-box model quite well,	18
3.5	Schematic model of a cylinder representing the load.	19
3.6	An example of the spool opening area as a function of the spool position in a directional control valve with a linearized version for each direction. The black line is the nonlinear area curve and the blue dashed line is the linearized area curve.	21
4.1	Block diagram of an LQR controller with integral action and back-calculating anti-windup.	28
4.2	Block diagram of the LQRI controller with back-calculating anti-windup for the software pressure feed reducer.	30
4.3	Block diagram of a general robust controller system.	31

4.4	Block diagram showing the generic plant P with weights added on inputs and outputs.	31
4.5	Block diagram of closed loop system used for designing a \mathcal{H}_∞ controller for the software pressure feed reducer.	33
4.6	General block diagram of closed loop system for two degrees of freedom \mathcal{H}_∞ Loop-Shaping controller.	34
4.7	Block diagram of closed loop system used for designing two degrees of freedom \mathcal{H}_∞ Loop-Shaping controller.	35
4.8	Bode plot of the open loop frequency response for $G_s(s)$ corresponding to workport A.	37
4.9	Block diagram for PID controller with back-calculation anti-windup.	42
4.10	Block diagram showing how the spool force $u(t)$ from a controller is used together with the input current, $i_o(t)$, from the operator.	43
5.1	Results from simulations without flow disturbance.	48
5.2	Results from simulations with a flow disturbance as a negative pressure derivative of -157 bar/s.	49
6.1	Block diagram showing the controller output conversion from Figure 4.10 together with the notch filter depicted in blue.	52
7.1	Schematic description of the boom lift and boom down test cases. 1. Boom down, 2. Boom lift. From [7]. Adapted with permission.	56
7.2	Behaviour from closing the valve as fast as possible 60 bar below the maximum pressure of 110 bar during a boom lift with 100 % operator signal.	58
7.3	Results from boom down test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.	60
7.4	Results from boom lift test case with 90 bar maximum pressure and 100 % operator signal. Notch filter enabled.	61
7.5	Results from boom lift test case with 110 bar maximum pressure and 100 % operator signal. Notch filter enabled.	62
7.6	Results from boom lift test case with 110 bar maximum pressure and 60 % operator signal. Notch filter enabled.	63
7.7	Results from digging test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.	64
7.8	Results from digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter disabled.	65
7.9	Oscillations from PID with bumpless transfer without the notch-filter. Boom lift test case with 110 bar maximum pressure and 100 % operator signal.	66
7.10	Dropping pressure from PID with bumpless transfer. Digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter disabled.	67
7.11	Results from PID with bumpless transfer using tunings from digging test case with added notch filter. Boom lift test case with 110 bar maximum pressure and 100 % operator signal.	68

7.12 Results from PID with bumpless transfer using tunings from boom lift test case. Digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter enabled. 69

List of Tables

1.1	Component specification for the system depicted in Figure 1.1.	6
5.1	Performance metrics from simulations without flow disturbance.	47
5.2	Performance metrics from simulations with flow disturbance.	47
7.1	Table showing the system limitations obtained by closing the valve as fast as possible. 60 % and 100 % are the operator signals, and 90 bar and 110 bar are the maximum pressures.	57
7.2	Performance metrics for boom down test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.	60
7.3	Performance metrics for boom lift test case with 90 bar maximum pressure and 100 % operator signal. Notch filter enabled.	61
7.4	Performance metrics for Boom lift test case with 110 bar maximum pressure and 100 % operator signal. Notch filter enabled.	62
7.5	Performance metrics for boom lift test case with 110 bar maximum pressure and 60 % operator signal. Notch filter enabled.	63
7.6	Performance metrics for digging test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.	64
7.7	Performance metrics for digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter disabled.	65

Nomenclature

Model Parameters

β	Hydraulic fluid bulk modulus	Pa
Δp_A	Workport A valve opening pressure difference	Pa
ρ	Hydraulic fluid density	kg/m ³
A, B, C, D	State space matrices	-
A_1	Cylinder chamber 1 area	m ²
A_2	Cylinder chamber 2 area	m ²
A_d, B_d, C_d, D_d	Discrete state space matrices	-
A_{cspc}	Pressure compensator cross section area	m ²
A_{cs}	Spool cross section area	m ²
C_q	Flow coefficient	-
c_s	Spool damping coefficient	Ns/m
c_{pc}	Pressure compensator damping coefficient	Ns/m
F_{pc0}	Pressure compensator spring force when in zero position	N
i_{min}	Minimum current required to open the valve	mA
k_s	Spool spring stiffness	N/m
k_{pc}	Pressure compensator spring stiffness	N/m
K_{soA}	Spool area function linear approximation slope	m
l_p	Cylinder stroke	m
m_s	Spool mass	kg
m_{pc}	Pressure compensator mass	kg
T_s	Sampling time	s
V_1	Cylinder chamber 1 volume	m ³
x_{dA}	Workport A spool deadband offset	m

Model Variables

$A_A(t)$	Workport A spool opening area	m^2
$A_B(t)$	Workport B spool opening area	m^2
$F_s(t)$	Spool force from pilot pressure	N
$F_{pc}(t)$	Pressure compensator force	N
$i(t)$	Pilot valve input current	mA
$i_o(t)$	Operator input	mA
$p(t)$	Pump pressure	Pa
$p_A(t)$	Workport A pressure	Pa
$p_B(t)$	Workport B pressure	Pa
$p_p(t)$	Pilot valve pressure	Pa
$p_T(t)$	Tank pressure	Pa
$p_{pc}(t)$	Pressure between spool and pressure compensator	Pa
$q_A(t)$	Flow out of workport A	m^3/s
$q_B(t)$	Flow into workport B	m^3/s
$u(t)$	Linear valve model input	N
$u_o(t)$	Linear valve model operator input	N
$x_p(t)$	Cylinder piston position	m
$x_s(t)$	Spool position	m
$x_{pc}(t)$	Pressure compensator position	m

LQR Parameters & Variables

J	Cost function	-
K	LQR gain	-
K_r	Reference tracking gain	-
Q	State weight matrix	-
R	Control input weight matrix	-

LQRI Parameters & Variables

J	Cost function	-
K	LQR gain	-
K_b	Anti-windup gain	-
K_r	Reference tracking gain	-

Q	State weight matrix	-
R	Control input weight matrix	-
T_d	Derivative time constant	s
T_i	Integrator time constant	s
T_{ri}	Integrator reset time	s
u_{LQR}	LQR control signal	N

\mathcal{H}_∞ Parameters & Variables

γ	Overbound and performance metric	-
d	Normalised flow disturbance	-
G_l	Load plant	-
G_v	Valve plant	-
K	\mathcal{H}_∞ controller	-
n	Normalised measurement noise	-
r	Normalised reference	-
$u_{o,norm}$	Normalised operator input	-
W_d	Flow disturbance weight	-
W_e	Reference tracking error weight	-
W_i	Reference weight	-
W_n	Measurement noise weight	-
W_r	Desired closed loop behaviour	-
W_u	Control input weight	-
W_{uo}	Operator input weight	-
y	Plant output	Pa
z_e	Reference tracking performance output	-
z_u	Control signal performance output	-

\mathcal{H}_∞ Loop-Shaping Parameters & Variables

β	Scaled reference	-
φ	Perturbation input	-
$\Delta_{M_s}, \Delta_{N_s}$	Uncertainty matrices	-
γ	Overbound and performance metric	-
\mathbf{e}	Reference tracking performance output	-

\mathbf{r}	Reference signal	-
\mathbf{u}_s	Control signal	-
\mathbf{y}	Plant output	-
ρ	Scalar tuning parameter	-
$G(s)$	Plant	-
$G_p(s)$	Perturbed plant	-
$G_s(s)$	Augmented plant	-
K	Controller state space system	-
M_s, N_s	Left coprime factorisation matrices	-
T_{ref}	Reference transfer function	-
W_1	Pre-compensator weight	-
W_i	Prefilter weight for model matching	-

MPC Parameters & Variables

ϵ	Soft state constraints slack variable	-
ρ_ϵ	Slack variable weight	-
\mathbf{u}_{max}	Control signal upper bound	-
\mathbf{u}_{min}	Control signal lower bound	-
\mathbf{x}_{max}	State upper bound	-
\mathbf{x}_{min}	State lower bound	-
K_{max}	Maximum solver iterations	-
N	Prediction horizon	-
Q	State weight matrix	-
Q_f	State weight matrix for end of prediction	-
R	Control input weight matrix	-
V_{max}^p	Constraint softness	-

PID Parameters & Variables

$e(t)$	Reference tracking error	-
$I[k]$	Approximation of integral part	-
K_b	Anti-windup gain	-
K_d	Derivative gain	-
K_i	Integral gain	-

K_p	Proportional gain	-
T_d	Derivative time constant	-
T_i	Integrator time constant	-
T_{pred}	Prediction time for enabling controller	-

Performance Metrics & Variables

e_{end}	Relative error at end of sequence	%
p_{end}	Pressure at end of sequence	bar
$p_{override}$	Pressure when the controller is enabled	bar
p_{peak}	Maximum pressure during sequence	bar
T_{final}	Time from enabling controller until 5 bar from maximum pressure	ms
$T_{overridden}$	Time that operator is overridden	ms

Other Parameters & Variables

ω_0	Notch frequency	rad/s
ω_c	Notch width	rad/s
$i_c(t)$	Controller current	mA
$N(s)$	Notch filter transfer function	-
p_{max}	Software pressure feed reducer maximum pressure	Pa
p_{pv}	Controller pilot pressure	bar
T_{stroke}	Minimum stroke time for boom cylinder on backhoe	s
V_{cyl}	Boom cylinder chamber volume on backhoe	m ³

1

Introduction

Today the field of mobile hydraulics is moving more and more towards implementation of different software functions, for example when it comes to controlling flows in an optimal way and reducing oscillations. Even though the mobile hydraulics industry is moving more towards electronic software solutions, there are still many pure hydromechanical components and system solutions on the market. Pure hydromechanical components are often nonflexible and not that easy to adapt to different applications and customise based on customer requirements. Therefore, manufacturers of mobile machines today requests more advanced software solutions from companies manufacturing hydraulic components and systems.

Using more software solutions makes it possible to replace hardware components, which may lead to reduced cost and reduced material consumption, something that is requested in today's society. It is also easier to develop new functionality for mobile machines using hydraulic components with a higher grade of software which may lead to an improved working environment for the operator and increased energy efficiency.

In mobile machines like backhoes, excavators and forestry machines the cranes are manoeuvred with hydraulic cylinders and these are in turn controlled by a component called directional control valve. This valve mainly controls the speed and direction of the crane's arms by controlling the hydraulic flow going into the hydraulic cylinders [1]. To limit the force obtained from the crane in order to not fatigue the material in the arms, the hydraulic output pressures from the valves and thus the pressures going into the hydraulic cylinders are limited. The pressure is limited by a component in the valve called pressure feed reducer which is completely hydromechanical and thus not possible to adjust during usage of the machine [1]. The functionality of the mobile machines previously mentioned can be improved if the hydraulic pressure from the directional control valve can be controlled during usage of the machines. For example, when digging using an excavator or a backhoe there might be fragile things in the ground, or the digging might be performed close to a wall. In such cases, it would be good to be able to lower the force from the crane in order to prevent the fragile objects and then be able to increase the force again when not digging close to fragile objects. This may be done by replacing the hydromechanical pressure feed reducer with a software solution using already existing components in the directional control valve.

The focus of this master's thesis has been to investigate which control algorithms that are suitable for pressure control when replacing the mechanical pressure feed reducer with a software solution. The work has been carried out at Parker Hannifin: Mobile Hydraulic Systems Division Europe located in Borås, Sweden. Parker provided all the necessary hardware to implement the software solution and also a backhoe used for testing.

1.1 Aim

This master's thesis work has aimed at evaluating different control algorithms used for pressure control when replacing the mechanical pressure feed reducer with a software solution using the existing components in a directional control valve. This means that the dynamics of the electrohydraulic directional control valve have been investigated and which model-based control strategies that are suitable for the application have been examined.

1.2 Problem Formulation

To be able to evaluate different control algorithms when replacing the mechanical pressure feed reducer with a software solution, and implement the algorithms on the available hardware, there are some different requirements that must be fulfilled. The output pressure from the directional control valve should be limited using the same signal and actuator as the operator of the mobile machine uses to control the flow out of the valve and thus the movement of the crane. This implies that the pressure feed reducer software function should override the operator when the output pressure from the valve is approaching the currently maximum allowed pressure. However, the function should not override the operator when the pressure is far away from the maximum pressure and the operator should not feel that the control of the machine is lost. Besides this, the only measurement available for the function is the output pressure from the valve. The software function will hereafter be called software pressure feed reducer.

The process of developing control algorithms for the software pressure feed reducer involves several different steps and therefore a model-based approach is used to accomplish the task. This means that the following steps are carried out:

- I. Analyse the directional control valve and build a mathematical model of the complete electro-hydraulic system.
- II. Synthesise different suitable model-based control algorithms.
- III. Simulate the synthesised controllers using the developed mathematical model.
- IV. Implement the controllers on the available hardware.
- V. Test and validate the software function on a backhoe.

The steps will be iterated in order to improve the final result and to get a good workflow.

1.3 Research Questions

The research questions that should be answered are formulated as:

- I. What is a suitable model of the electrohydraulic directional control valve when designing a software pressure feed reducer, and which parts of the system should be included in it?
- II. Which model-based control algorithm is most suitable for a software pressure feed reducer in an electrohydraulic directional control valve when analysing oscillations, rise time, steady state error and overshoot compared to the mechanical pressure feed reducer?
- III. Are there any limitations in the physical system when using model-based control algorithms for controlling the output pressure of an electrohydraulic valve?

1.4 Problem Discussion

When replacing the mechanical pressure feed reducer with a software solution in an existing directional control valve there are difficulties and properties that must be considered. Even though a software solution yields more flexibility and can introduce new functionalities, there are some things that might be problematic and thus must be considered.

The directional control valve can be used on a variety of different mobile machines and thus must function regardless of the application. This means that the pressure feed reducer also must work properly no matter which application the valve is used for. This is a challenge for a software solution since different applications affects the behaviour of the hydraulic system and the pressure quite a lot. Of course this affects the mechanical pressure feed reducer as well, but it is robust and handles different valve applications well.

Another thing that is important to consider when implementing the pressure feed reducer function is the operator experience. The function should not limit the operator more than necessary which means that the operator signal should not be overridden far away from the maximum allowed pressure. If not, the hydraulic function on the mobile machine will feel slow and unresponsive for the operator which leads to a worse working environment.

Besides this, the demand on hydraulic components and functions in hydraulic systems are high for mobile machines since downtime is expensive. This means that the software pressure feed reducer must be reliable and should not affect the overall performance of the valve.

1.5 Scope & Boundaries

The scope of this master's thesis is to evaluate different control algorithms that can be used to replace the mechanical pressure feed reducer in a directional control valve with a software solution rather than developing a complete product. This means that more focus will be on comparing different control algorithms and analysing the results. In order to carry out the master's thesis successfully the boundaries below have been set up.

The hardware available for implementation of the software pressure feed reducer is an electrohydraulic directional control valve of the type K170LS from Parker Hannifin mounted on a backhoe of the make Huddig from 1998, equipped with modern electronics and hydraulics from Parker. This means that model parameters will be adapted to this hardware and the final tests and verifications will be performed with this hardware. The analysis and validation of the results will be based on implementation on the boom function of this backhoe.

When modelling the directional control valve, the hydraulic flow is assumed to be laminar and forces due to the hydraulic flow will be neglected. Also pressure drops in channels in the valve are neglected.

1.6 Previous Works

Hydraulic systems is a field where an extensive amount of research has been carried out during the years. However, the research results are not commonly used in practice. One part of the field that is heavily researched is how to model different types of directional control valves. An example of this is [2] where the models are also analysed with a focus on dynamic characteristics and energy efficiency.

Multiple approaches for controlling the position of hydraulic cylinders have previously been developed and tested, for example in [3] a robust controller for trajectory tracking of an excavator crane was developed using μ -synthesis in order to handle nonlinearities and disturbances. The developed controller was tested on an excavator with satisfactory results. Besides this, in [4] a model predictive controller was designed for position control of a cylinder. The controller was, however, only tested in simulation, but with satisfactory results. Even though both these approaches are for controlling the position of a cylinder, rather than the pressure focused on here, they can be useful. This is because they show what kind of control algorithms have been successfully applied to hydraulic systems.

Controlling pressure electrohydraulically with directional control valves has not been done extensively. However, some research has been performed on the subject. In [5] several different PID tuning rules based on a model of a directional control valve attached to a cylinder pushing on a stiff surface are presented. The PID controllers were tested experimentally and produced satisfactory results. They were, however, only tested on a cylinder pushing on a stiff surface, not on a moving cylinder or on a mobile machine. Moreover, in [6] a valve specifically designed for pressure control is

developed together with a corresponding closed loop controller in order to improve performance compared to traditional open loop pressure control valves. While this work is not related to directional control valves it shows that pressure control within hydraulics is a topic that still needs research and development.

With the previous work mentioned in mind, investigating different techniques for controlling pressure using a directional control valve in mobile machinery is an interesting research topic. It is especially interesting to investigate which control algorithm works best when applied to a mobile machine.

1.7 System Overview

The system used for testing and implementation of the software pressure feed reducer can be seen in Figure 1.1, where the components depicted are listed in Table 1.1. The hydraulic cylinder on the backhoe controls the boom function and is connected to the directional control valve with two hydraulic hoses. The pressure in each hose is measured using two pressure sensors that are connected to a controller module using analog signals depicted as red lines in the figure. The software pressure feed reducer is to be implemented on the controller module together with the already implemented software for controlling the boom function on the backhoe. The directional control valve is also electrically connected to the controller module since the valve and thus the hydraulic flow into the cylinder is manoeuvred by the controller module.

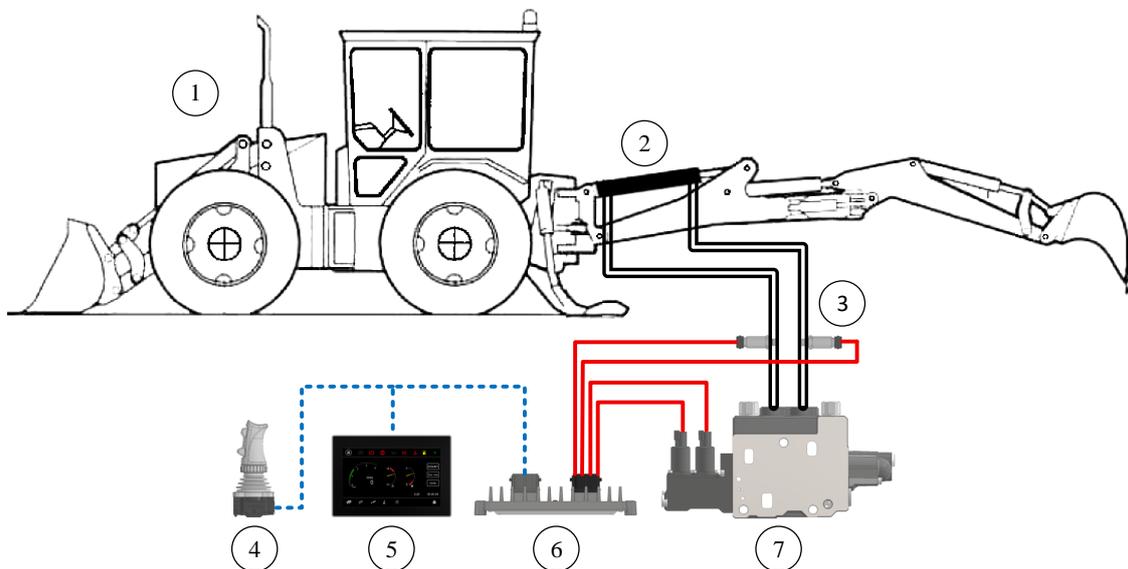


Figure 1.1: Schematic overview of the system used for implementation and tests. 1. Backhoe, 2. Boom cylinder, 3. Pressure sensors, 4. Joystick, 5. Touch display, 6. Controller module, 7. Directional control valve. From [7]–[12]. Adapted with permission.

Table 1.1: Component specification for the system depicted in Figure 1.1.

	Description	Component
1	Backhoe	Huddig 1160 1998 [7]
2	Boom cylinder	Huddig 1160 original boom cylinder [7]
3	Pressure sensors	Parker SensoControl SCP [8]
4	Joystick	Parker IQAN-LC6-X05 [9]
5	Touch display	Parker IQAN-MD4-7 [10]
6	Controller module	Parker IQAN-MC43 [11]
7	Directional control valve	Parker K170LS [12]

To the controller module a touch display and a joystick is connected via a CAN-bus, illustrated with the blue dashed line in Figure 1.1. The joystick is used by the operator to control the hydraulic flow from the directional control valve to the hydraulic cylinder, i.e. the movement of the hydraulic cylinder. The touch display is used to easily adjust tuning parameters corresponding to the software in the controller module.

2

Directional Control Valve

In order to grasp the concept of the models and control algorithms presented in this thesis it is necessary to know what a directional control valve is and how it works. In this chapter the basics of a directional control valve are explained and especially the function of the mechanical pressure feed reducer are presented in depth.

Simply explained, a directional control valve is used to control the movement of the piston in a hydraulic cylinder within a variety of different applications. Mobile machines are an application where directional control valves are commonly used to control cylinders mounted on different joints and cranes. Due to the high number of different mobile machines there are many different types of directional control valves. They can, for example, be actuated using mechanical actuators, pneumatics or hydraulic pilot pressures [1]. The type of valve described in this chapter is actuated using electrohydraulic proportional pilot control and is used to control a double-acting cylinder. Besides this there are many different parameters and specifications that can vary between valves of the same type which makes each valve almost unique. Therefore, details about the specific electrohydraulic directional control valve used for implementation in this thesis are presented.

2.1 Basic Functionality

A directional control valve consists of a spool that is located in a valve body and it can be moved in two different directions. The spool is used to control the amount of hydraulic flow going out of the valve and when the spool is moved in different directions oil flow is directed between different ports. A hydraulic scheme of an electrohydraulic directional control valve connected to a pump and tank can be seen in Figure 2.1. When the valve is in rest and no force is applied to the ends of the spool it is located in its centre position. The valve depicted in Figure 2.1 has a so called closed-centre spool which means that in the centre position no oil can flow between the pump and the two workports A and B and between the two workports and the tank [1]. A workport is the connection on the valve that should be attached to the corresponding port on the hydraulic cylinder, the A port on the valve should be connected to one end of a hydraulic cylinder and the B port to the other end of the cylinder. When the two workports are closed in the centre position the attached cylinder will not move and the valve is said to be closed.

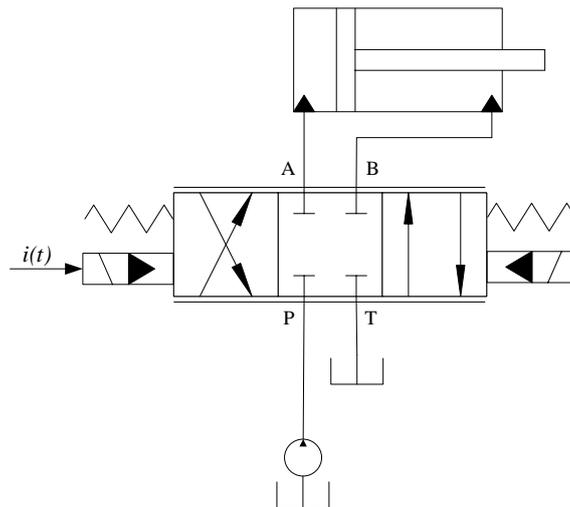


Figure 2.1: Hydraulic scheme for a directional control valve with a closed-centre spool that is electrohydraulic pilot actuated and connected to a pump, tank and double acting cylinder. A and B are the two workports, P is the hydraulic pump and T is the tank.

If a force is applied on the right end of the spool it will start to move to the left and oil will start to flow from the pump to workport A and from workport B to the tank [1]. This can be illustrated by the right box changing place with the middle box in the scheme in Figure 2.1 and the flow directions are shown with the arrows. If, on the other hand, a force is applied to the left end of the spool it will move to the right and oil will flow from the pump to workport B and from workport A to the tank.

When the spool starts to move from the centre position to either the left or right, there is a deadband which means that the spool has to be moved a certain distance before any oil can start to flow from the pump to one of the workports and from the other workport [1]. Once the spool has been moved past the deadband the flow will start to increase due to the valve opening increasing with the moved distance. How the flow is increasing depends on the pressure-drop over the spool opening and on the spool opening area as a function of the spool position. This function can vary a lot between different valves and different spools, but an example of the relationship between the spool opening area and the spool position can be seen in Figure 2.2.

In an electrohydraulic pilot actuated valve the spool is moved using hydraulic pilot pressure that gives rise to a force at the end of the spool [1]. The pilot pressure is controlled using a pilot valve which, simply explained, consists of proportional solenoid and a separate hydraulic valve. The pilot valve in turn is controlled by a current, which is illustrated with $i(t)$ in Figure 2.1.

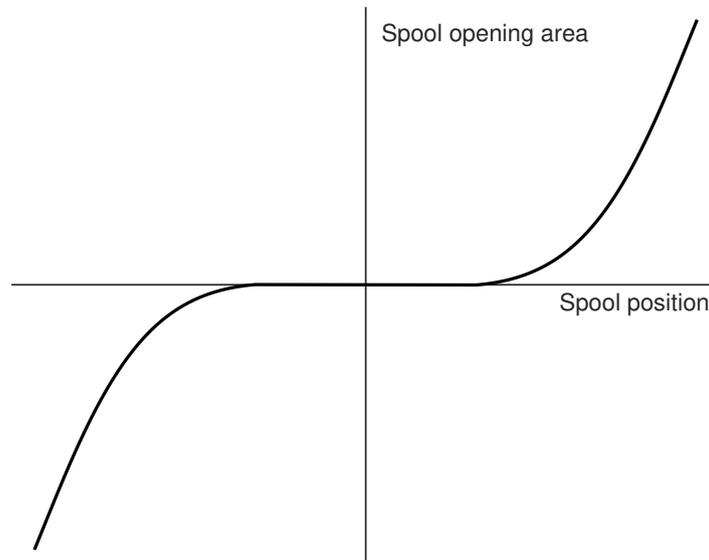


Figure 2.2: An example of the spool opening area as a function of the spool position in a directional control valve. Note the deadband and that the function is non-linear.

2.2 Pressure Compensator

A pressure compensator is a mechanical component that can be used in directional control valves to get a constant hydraulic flow rate for a certain input current, independent of the load and pump pressure [12]. The compensator is located before the main spool such that the hydraulic oil from the pump flows via the compensator to the spool.

The pressure compensator achieves a constant hydraulic flow rate for a certain input current by reducing the incoming pump pressure when the load pressure increases such that the pressure drop across the main spool is constant [1]. This is done by a mass that is balanced between the load pressure and the pressure in the channel between the compensator and main spool. This mass has a similar functionality as the main spool in the valve and adjusts the opening area between the pump and the spool and thus the flow into the spool is adjusted. A spring is also attached to this mass on the same side as the load pressure which is illustrated in a hydraulic scheme in Figure 2.3. The pressure compensator block in the figure represents the mass, and the dashed lines represent channels that directs the pressures used to balance it. The load pressure, or the pressure in the workport, is directed to the compensator via the main spool through a narrow channel and is called the load sensing (LS) signal [1]. The flow in the LS-signal is very low and can be assumed to not affect the flow in the workport. If the pressure in the channel between the

pressure compensator and the main spool overcomes the spring force and the LS signal it closes the compensator, and therefore reduces the pump pressure before it reaches the main spool. In this way the pressure drop over the main spool is kept to a constant pressure level which depends on the stiffness of the spring used.

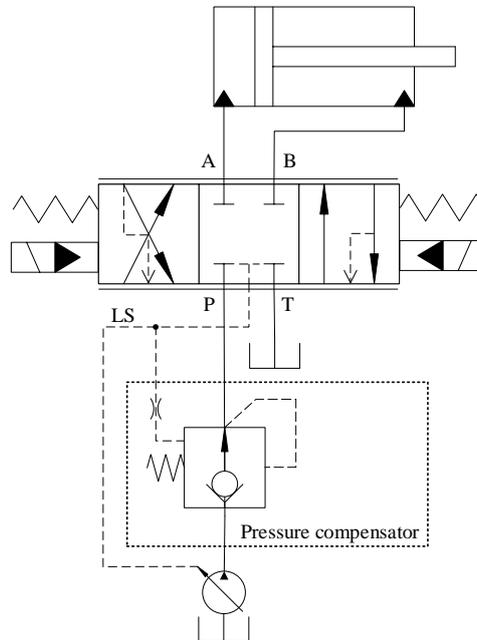


Figure 2.3: A simplified hydraulic scheme of a directional control valve with a pressure compensator.

2.3 Mechanical Pressure Feed Reducer

A pressure feed reducer is a mechanical component used to limit the maximum pressure in the workports of the directional control valve to protect the equipment it is controlling [1]. There is one feed reducer for each workport and the maximum pressure can be set individually for each port and on each valve to get different maximum pressures for individual functions.

The pressure feed reducer senses the pressure in the workport using the LS-signal described in Section 2.2 and contains a spring which is compressed when the sensed hydraulic pressure increases [1]. When the maximum pressure allowed in the workport is reached the spring is compressed enough to let the LS-signal flow directly to the tank instead of flowing to the pressure compensator. This means that the compensator closes and no hydraulic fluid flows into the workport and thus, the pressure in the workport stops increasing.

The maximum pressure is adjusted by changing the preload of the spring in the feed reducer [1]. This is something which is done in production as a part off the manufacturing process of the directional control valve and once the valve is installed on a mobile machine the maximum pressure is usually not adjusted.

To summarise, the mechanical pressure feed reducer limits the maximum pressure by limiting the hydraulic flow from the pump into the workport which means that it does not work as an ordinary pressure relief valve attached to the workport. It can for example not reduce the pressure in the workport if it has already reached a too high level.

2.4 The K170LS Valve Used for Implementation

The valve that is used in this thesis is a K170LS valve, which is a directional control valve. A cross section view of it can be seen in Figure 2.4 where the two workports can be seen on top and the different components of the valve are labelled.

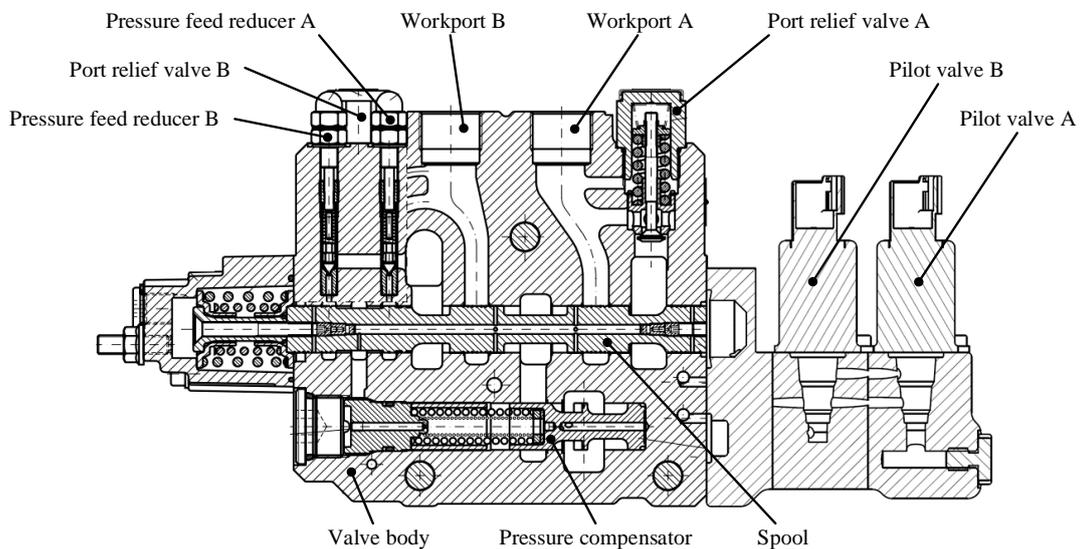


Figure 2.4: Cross section view of a K170LS directional control valve where the different components are labelled. From [12]. Adapted with permission.

The K170LS is a load sensing valve that can be built up with multiple sections, where each section is a directional control valve described in Section 2.1 used to control one cylinder. Each section can be equipped with its own pressure compensator, making the sections independent of each other. The load sensing functionality measures the pressure in the workports for each section and controls the pump to give adequate pressure for the current load [1]. The pressure compensator in each section then regulates the pressure to the spool in that section to yield a constant pressure-drop over the spool opening. This implies that in theory, the flow in the sections with compensators is proportional to the open spool area [1].

The pressure feed reducers on the K170LS valve are adjustable between 30 and 330 bar maximum pressure and for high pressure settings and high hydraulic flows the mechanical pressure feed reducer starts to limit the pressure approximately 30 bar below the maximum pressure [12].

The specific valve section that is used for implementation in this thesis is equipped with a closed-centre spool for a double-acting cylinder. The spool is designed such that the flow out of workport A is approximately four to five times higher than the flow out of workport B. Besides this the spool opens the workport to the tank slightly before the other workport opens to the pump. Further, the valve section has a pressure compensator, which together with the load sensing gives a constant flow rate for a certain spool position as explained previously. It is also equipped with pressure feed reducers for both workports. Apart from the pressure feed reducers on the workports there are also port relief valves on both workports. These have the functionality to protect the valve from high pressure peaks generated from the load [1]. The spool in the valve section is electrohydraulically actuated using a pilot pressure, which is controlled by a current.

3

Mathematical Modelling

The control algorithms presented in this thesis are model-based control algorithms and it is therefore necessary to have a mathematical model of the complete electro-hydraulic system from input to output. An overview of the complete system, including the load, in form of a block-diagram can be seen in Figure 3.1. The input is the current $i(t)$ to the pilot valve controlling the pilot pressure $p_p(t)$ for one end of the spool in the directional control valve. The pump pressure $p(t)$ is modelled as a disturbance input since it affects the behaviour of the system, but it cannot be used as a control input since it cannot be changed by the control system. The pump pressure goes into the pressure compensator and the output from this block is the pressure in the channel between the pressure compensator and the spool in the directional control valve, shown as $p_{pc}(t)$ in Figure 3.1.

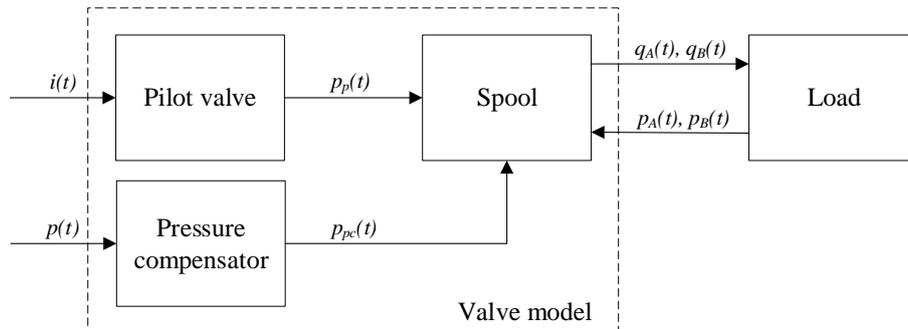


Figure 3.1: Overview of the complete system to model and its different parts.

The output from the valve model is the flow in the workports, represented by $q_A(t)$ and $q_B(t)$ in Figure 3.1, which results in the pressures $p_A(t)$ and $p_B(t)$ due to the connected load. Since a double acting cylinder will be used for implementation two different scenarios have to be modelled. One scenario is when the valve is opened such that the flow is directed from the pump to workport A and from workport B to the tank. The second scenario is when the valve is opened in the opposite direction such that the flow is directed from the pump to workport B and from workport A to the tank. However, the equations for the two different scenarios are the same, but with different parameters, and thus only equations for the case when oil flows from the pump to workport A are presented in this chapter. This means that $q_A(t)$ is the flow out from workport A, and $q_B(t)$ is the flow back into workport B.

In this chapter the different parts of the model in Figure 3.1, including the load connected to the valve, are derived. The model derived is a simplified model of the system that does not take flow forces and pressure drops in channels in the valve into account, since that is outside the scope of the thesis project, as mentioned in Section 1.5.

In the end the model is formulated on state space form in order to be able to apply model-based control algorithms. In addition to this, verification of a part of the model is described.

3.1 Valve Model

In this section the directional control valve described in Chapter 2 is modelled. The inputs to this model are the control signal $i(t)$, i.e. current to the pilot valve, and the input disturbance $p(t)$ which is the pump pressure into the pressure compensator. The output is the hydraulic flow out of workport A, $q_A(t)$. The pilot valve, the spool and the pressure compensator will be modelled separately and then put together using flow equations.

3.1.1 Pilot Valve

The input signal to the valve model is controlling the pilot pressure through the pilot valve using a current. According to knowledge from Parker Hannifin the dynamics of this pilot valve is significantly faster than the main spool of the directional control valve. Therefore the dynamics of the pilot valve is not modelled and instead a lookup-table is used to map the input current to the pilot pressure, i.e. $p_p(i(t))$. This mapping is not linear which means that there is a static non-linearity between the input current and the pilot pressure.

The pilot pressure is converted into a force on the spool, $F_s(t)$ according to

$$F_s(t) = A_{cs}p_p(i(t)) \quad (3.1)$$

where A_{cs} is the cross section area of the spool the pilot pressure is acting on.

3.1.2 Spool

A simplified schematic model of the spool can be seen in Figure 3.2, where $x_s(t)$ is the spool position measured from the position where the valve starts to open, m_s is the mass of the spool, k_s is the spring constant of the spring that is keeping the spool in the centre position and c_s is the damping coefficient caused by the hydraulic oil acting on the spool.

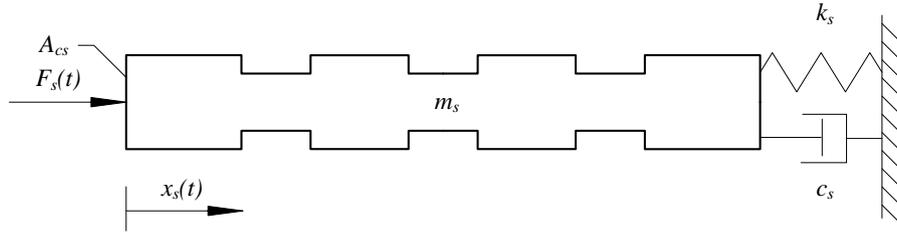


Figure 3.2: Schematic model of the spool in the directional control valve used to describe its dynamics.

The equation of motion for the spool can be obtained from Newtons Second Law, yielding

$$m_s \ddot{x}_s = F_s(t) - k_s(x_s(t) + x_{dA}) - c_s \dot{x}_s \quad (3.2)$$

where x_{dA} represents the deadband, i.e. the distance from the centre position of the spool to the position where the valve starts to open.

From the valve position, the opening areas of the valve can be found from nonlinear area-curves specific to the spool used, similar to the one seen in Figure 2.2. One opening area, $A_A(x_s(t))$, is from the hydraulic pump to workport A and one opening area, $A_B(x_s(t))$, is from workport B to the tank.

3.1.3 Pressure Compensator

A schematic view of the pressure compensator can be seen in Figure 3.3 where A_{cspc} is the cross section area of the compensator, $F_{pc}(t)$ is an external force acting on the compensator from the hydraulic pressures, m_{pc} is the mass of the compensator, $x_{pc}(t)$ is the position of the compensator measured from the point where it starts to open, k_{pc} is the spring stiffness of the spring that closes the compensator and c_{pc} is the damping coefficient for the compensator caused by the hydraulic fluid.

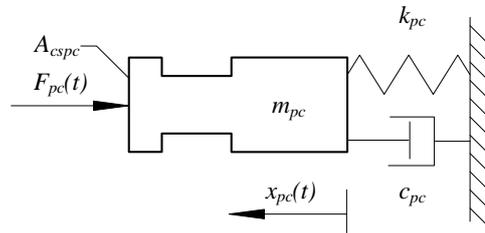


Figure 3.3: Schematic model of the pressure compensator in the directional control valve used to describe its dynamics.

From Figure 3.3 the equation of motion for the pressure compensator can be derived using Newtons Second Law as

$$m_{pc} \ddot{x}_{pc} = F_{pc0} - k_{pc} x_{pc}(t) - c_{pc} \dot{x}_{pc} - F_{pc}(t) \quad (3.3)$$

where F_{pc0} is the force from the spring when the compensator is in the zero position. The force $F_{pc}(t)$ comes from the pressure in the workport, $p_A(t)$, and the pressure in the channel between the pressure compensator and the spool, $p_{pc}(t)$. The pressure in the workport acts in the positive $x_{pc}(t)$ -direction, and the pressure in the channel between the pressure compensator and the spool is acting in the negative $x_{pc}(t)$ -direction. Therefore, the force acting on the compensator becomes

$$F_{pc}(t) = A_{cspc}p_A(t) - A_{cspc}p_{pc}(t). \quad (3.4)$$

The opening area of the pressure compensator between the pump pressure and the pressure in the channel between the compensator and spool can then be found from a non-linear area curve $A_{pc}(x_{pc}(t))$.

3.1.4 Hydraulic Flow

Hydraulic flow q through an orifice can in general be described by

$$q = C_q A \sqrt{\frac{2}{\rho} \Delta p} \quad (3.5)$$

where C_q is the flow coefficient, A is the orifice area, ρ is the density of the fluid and Δp is the pressure drop over the orifice [13]. Applying this equation to the open areas of the spool valve yields

$$q_A(t) = C_q A_A(x_s(t)) \sqrt{\frac{2}{\rho} (p_{pc}(t) - p_A(t))} \quad (3.6a)$$

$$q_B(t) = C_q A_B(x_s(t)) \sqrt{\frac{2}{\rho} (p_B(t) - p_T(t))} \quad (3.6b)$$

where $p_{pc}(t)$ is the pressure in the channel between the pressure compensator and the spool and $p_T(t)$ is the tank pressure. The flow through the spool orifice is the same as the flow in the workport since it is assumed that there is no leakage. However, in practice there is a leakage between the spool and the valve body, but it is very small and can therefore be neglected. This also means that the flow in the pressure compensator orifice is the same as the flow in the spool orifice and out of workport A. Thus, applying the equation for flow through an orifice to the open area of the compensator yields

$$q_A(t) = C_q A_{pc}(x_{pc}(t)) \sqrt{\frac{2}{\rho} (p(t) - p_{pc}(t))}. \quad (3.7)$$

In order to use (3.7) in the final model of the valve used for control, the pump pressure has to be known, but as mentioned in Section 1.7 the pump pressure is not measured. This means that if the equation should be used the pump pressure has to be estimated or assumed constant. Since the hydraulic pump in the backhoe used for implementation is variable and gives different pressures depending on the states of all hydraulic functions on the machine it is not reasonable to assume the pump pressure

to be constant. Instead of trying to estimate the pump pressure and use (3.7), one approach is to assume that the pressure drop $p_{pc}(t) - p_A(t)$ over the spool opening area $A_A(x_s(t))$ in (3.6) is constant. This is a quite large simplification, but since the function of the pressure compensator is to keep the pressure drop over the spool opening constant at a specific level, this is reasonable. Besides this, the dynamics of the load attached to the valve will affect the pressure in the workport more than the pressure compensator and, thus, the dynamics of the pressure compensator can be neglected. For the directional control valve used for implementation, K170LS, the pressure drop over the spool opening that the compensator tries to keep is known.

Assuming that the tank pressure is zero together with the assumption above, the flow expressions in (3.6) can be simplified to

$$q_A(t) = C_q A_A(x_s(t)) \sqrt{\frac{2}{\rho} \Delta p_A} \quad (3.8a)$$

$$q_B(t) = C_q A_B(x_s(t)) \sqrt{\frac{2}{\rho} p_B(t)} \quad (3.8b)$$

where Δp_A is the pressure drop over the spool opening area for workport A.

These equations together with the equation of motion for the spool in (3.2) describe the dynamics of the valve, from the input current to the pilot valve to the output flow of the valve.

3.1.5 Verification of Spool Model

All parts of the models developed in this chapter were implemented and simulated using MATLAB and Simulink, which means that the spool model derived in Section 3.1.2 can be verified thanks to a black-box model in Simulink available from Parker Hannifin. The black-box model is a model of the K170LS valve that will be used for verification and contains all details from the pilot pressure $p_p(t)$ as input to the spool position $x_s(t)$ as output. The model is developed in the simulation software Amesim and due to this there is no mathematical expression for the model available and it can therefore not be used as a part of the system model. Besides this the black-box model is verified against a physical valve which makes it suitable for verification of the valve model.

The black-box model is not only suitable for verifying the final valve model from pilot pressure to spool position, it is also helpful in finding the unknown damping coefficient c_s for the valve. This means that the black-box model was used to both tune the damping coefficient in the developed valve and for verification of the model. This was done by adding a step on the pilot pressure and simulating the two models in parallel and comparing the obtained spool positions. The final result can be seen in Figure 3.4 where two different pilot pressure steps are plotted, one at 15 bar and one at 8 bar. As can be seen the developed valve model does not follow the verified black-box model perfectly, but the overall behaviour is correct and the deviations are quite small. Despite the deviations the developed model is assumed to be accurate enough for the purpose in this thesis since more simplifications will be introduced in other parts of the model.

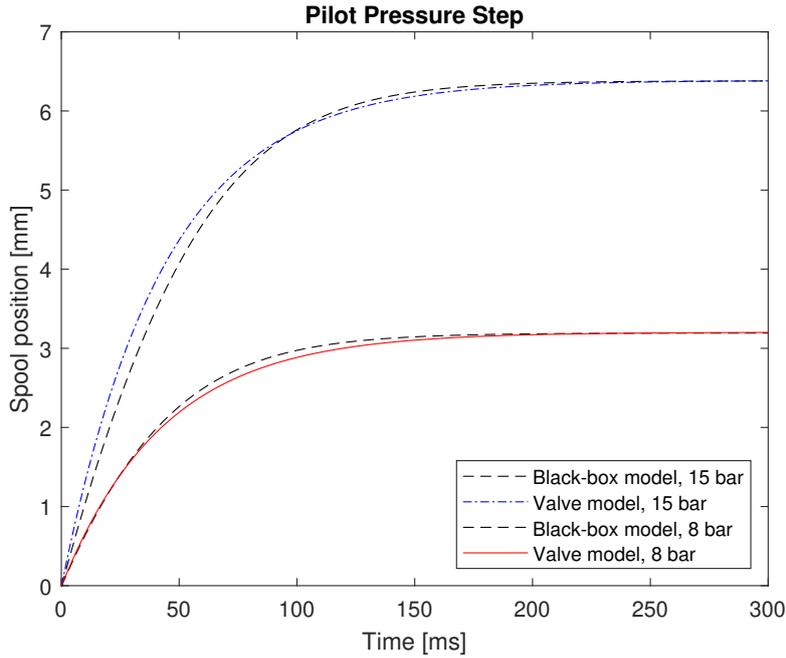


Figure 3.4: Comparison of valve model and black-box model for two different reference steps in pilot pressure. As can be seen, the valve model follows the verified black-box model quite well,

3.2 Load Model

The load that is connected to the valve determines the relationship between the flow coming from the valve and the pressure in the workports. In most applications of directional control valves, the load is a cylinder that controls a crane function on a mobile machine. For the application in this thesis, the load is a double-acting cylinder on the boom function of a backhoe. Since the dynamics of the crane on the backhoe changes significantly during different operations, for example digging and lifting objects, and when moving other axes on the crane, the crane is not modelled in detail since it would only be accurate for some specific cases. Instead, the load is modelled as a double-acting cylinder where the position and velocity of it is considered to be disturbances. A schematic model of such a cylinder can be seen in Figure 3.5, where $x_p(t)$ is the position of the piston in the cylinder, A_1 and A_2 are the areas the pressures act on and l_p is the stroke of the cylinder.

In general, the hydraulic flow into a volume of changing size caused by an inlet pressure p can be calculated as

$$q = \dot{V} + \frac{V}{\beta} \dot{p} \quad (3.9)$$

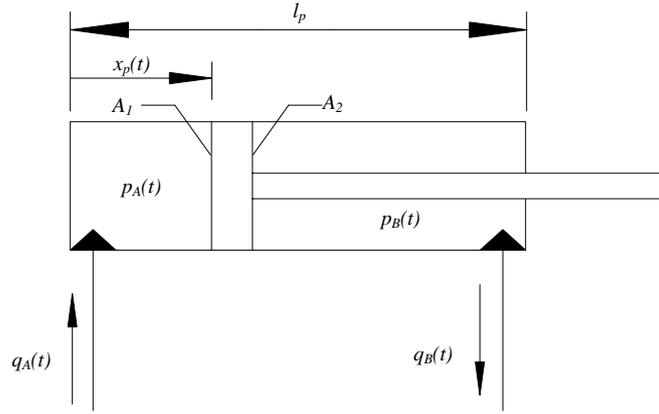


Figure 3.5: Schematic model of a cylinder representing the load.

where V is the volume and β is the bulk modulus [13]. The bulk modulus is a measure of how much pressure it takes to compress a fluid. Applying (3.9) to the two chambers in the hydraulic cylinder in Figure 3.5 yields

$$q_A(t) = A_1 \dot{x}_p + \frac{A_1 x_p(t)}{\beta} \dot{p}_A \quad (3.10a)$$

$$q_B(t) = A_2 \dot{x}_p - \frac{A_2 (l_p - x_p(t))}{\beta} \dot{p}_B. \quad (3.10b)$$

For the application in this thesis, the position of the cylinder is not measured. By assuming that the cylinder is at a fixed position, with zero velocity, the load can be simplified to a chamber of oil with a fixed volume. Even though the cylinder is not at a fixed position in this application, it can be a reasonable approximation. The reason for this is that the control will only be active when the pressure in the workport is close to the maximum pressure. In most cases when this happens, the cylinder is either moving slowly, or in some cases not at all. Using this approximation, the flow out of workport A can then be simplified to

$$q_A(t) = \frac{V_1}{\beta} \dot{p}_A \quad (3.11)$$

where V_1 is the volume of the chamber of oil. This volume represents the volume of oil in the cylinder, and it is unknown since the cylinder moves during operation. However, it represents the volume of oil in the cylinder, but it can also represent non-stiff behaviour of the backhoe. This can be seen from (3.11) where it can be noted that increasing V_1 makes the pressure increase slower for a given flow.

Thus a simplified load model as a relationship between the flow from the valve and the pressure in workport A has been obtained in (3.11). Workport B is omitted in the simplified load model since it is not of interest for the application when flow is directed to workport A.

3.3 State Space Representation

The models developed for the valve and the load in Section 3.1 and 3.2 respectively can be reformulated and written on state space form, which is necessary for applying model-based control algorithms. Choosing the states according to

$$\mathbf{x} = \begin{bmatrix} x_s \\ \dot{x}_s \\ p_A \end{bmatrix} \quad (3.12)$$

and combining (3.2), (3.8a) and (3.11) yields the state space representation

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_s \\ \ddot{x}_s \\ \dot{p}_A \end{bmatrix} = \begin{bmatrix} \dot{x}_s \\ \frac{1}{m_s} (F_s(t) - k_s(x_s(t) + x_{da}) - c_s \dot{x}_s) \\ \frac{\beta C_q A_A(x_s(t))}{V_1} \sqrt{\frac{2}{\rho} \Delta p_A} \end{bmatrix}, \quad (3.13)$$

where $F_s(t)$ is seen as the control input to the system. In the physical system the input is the current to the pilot valve, but since the conversion between the current and the force is only a static mapping the force can be seen as input instead, without changing the dynamics of the system. Using the spool force $F_s(t)$ as input instead of the input current $i(t)$ also results in that the non-linearity between the input current and the pilot pressure, i.e. the force on the spool, is removed.

3.4 Linear Model for Control

In order to develop model-based and state feedback control algorithms the states in the state space model has to be known, either by measuring them or by estimation. This means that to be able to use the state space model given in (3.13) the spool position $x_s(t)$, the spool velocity $\dot{x}_s(t)$ and the workport pressure $p_A(t)$ has to be known. As already mentioned the pressure will be measured, but there is no measurement available for the spool position, which requires this to be estimated in order to use the model on the form given in (3.13). However, this model can be rewritten to use the states according to

$$\mathbf{x} = \begin{bmatrix} p_A \\ \dot{p}_A \\ \ddot{p}_A \end{bmatrix} \quad (3.14)$$

instead of the states in (3.12) in order to remove the need of state estimation. This reformulation requires that (3.8a) is linearized, which can be done by approximating the spool opening area function $A_A(x_s(t))$ with a linear function on the form $K_{soA}x_s(t)$. The linearization is only valid when the valve is opened, which can be seen in Figure 3.6. After linearizing the area curve, (3.8a) can be written as

$$q_A(t) = C_q K_{soA} x_s(t) \sqrt{\frac{2}{\rho} \Delta p_A} \quad (3.15)$$

and when combining this with (3.11) the following expression for $x_s(t)$ is obtained:

$$x_s(t) = \frac{V_1 \dot{p}_A}{\beta C_q K_{soA} \sqrt{\frac{2}{\rho} \Delta p_A}}. \quad (3.16)$$

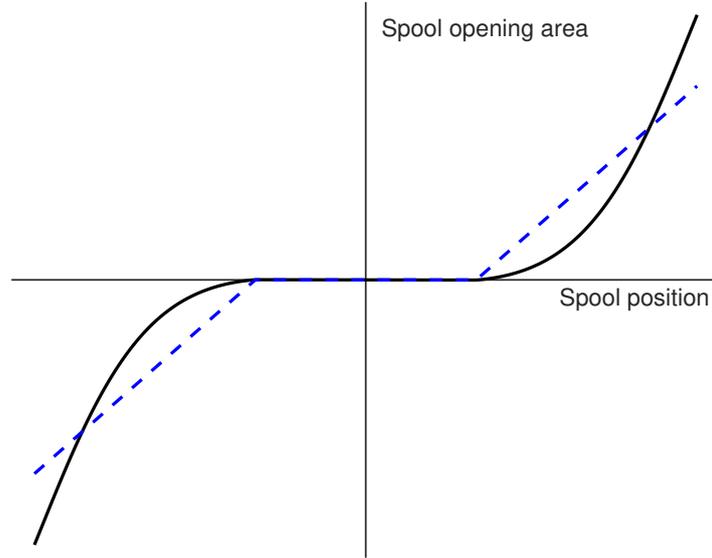


Figure 3.6: An example of the spool opening area as a function of the spool position in a directional control valve with a linearized version for each direction. The black line is the nonlinear area curve and the blue dashed line is the linearized area curve.

Using the expression in (3.16) together with (3.2) the state space model in (3.13) can be reformulated as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p}_A \\ \ddot{p}_A \\ \ddot{p}_A \end{bmatrix} = \begin{bmatrix} \dot{p}_A \\ \ddot{p}_A \\ \frac{\beta C_q K_{soA} \sqrt{\frac{2}{\rho} \Delta p_A}}{m_s V_1} F_s(t) - \frac{k_s}{m_s} \dot{p}_A - \frac{k_s \beta C_q K_{soA} \sqrt{\frac{2}{\rho} \Delta p_A}}{m_s V_1} x_{dA} - \frac{c_s}{m_s} \ddot{p}_A \end{bmatrix}. \quad (3.17)$$

With this state space model there is no need of estimating any states since the pressure p_A is measured, but some filtering may be required due to noise in the measurement signal. In order to facilitate the process of developing control algorithms for the software pressure feed reducer the model in (3.17) can be written on

3. Mathematical Modelling

a form that is more suitable for control according to

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{k_s}{m_s} & -\frac{c_s}{m_s} \end{bmatrix}}_A \mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{\beta C_q K_{soA} \sqrt{\frac{2}{\rho} \Delta p_A}}{m_s V_1} \end{bmatrix}}_B \underbrace{(F_s(t) - k_s x_{dA})}_{u(t)} \quad (3.18a)$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_C \mathbf{x} \quad (3.18b)$$

where the input $u(t)$ is the force on the spool compensated for the deadband offset x_{dA} and the output is the pressure p_A in the workport that should be controlled.

4

Control Algorithms

The theory behind the different control algorithms that should be designed for pressure control and evaluated for the software pressure feed reducer are presented in this chapter. The controllers described are linear quadratic regulators (LQR) with and without integral action, an \mathcal{H}_∞ controller, an \mathcal{H}_∞ loop shaping controller, a Model Predictive Controller (MPC) and a PID controller with bumpless transfer.

Since the controllers will be implemented on discrete hardware, discrete time controllers are designed. Also, the control algorithms are developed for the state space model in Section 3.4, meaning that the output from the controllers is a spool force offset expressed in newton. This force then needs to be converted into a current that can be applied to the directional control valve. This process is the same for all controllers, and is described at the end of the chapter.

Since the software pressure feed reducer is only supposed to limit the maximum pressure, and not actively drive the pressure to a set point, the controllers are not allowed to give a higher input signal than the operator. This means that the spool should only be closed in a satisfactory manner and not moved beyond the centre-position. This implies that the controller cannot actively reduce the pressure, and therefore it is important that the controlled pressure does not overshoot. Thus, when tuning the different controllers, they must be tuned to get a closed loop behaviour that fulfils this.

The controllers have been implemented using MATLAB and Simulink.

4.1 Discretisation

The final software pressure feed reducer was implemented on a discrete hardware with a fixed sampling time T_s of 10 ms and, thus, all controllers have to be in discrete time. This means that the continuous model developed in Chapter 3 must be discretised in order to apply the discrete model-based controllers. Using zero order hold, each value at a specific sample is constant until next sample which means that the discrete version of a continuous signal $\mathbf{u}(t)$ can be written as

$$\mathbf{u}[k] = \mathbf{u}(kT_s) = \mathbf{u}(t), \quad kT_s \leq t < (k+1)T_s \quad (4.1)$$

where k is the current sample. Assuming a sampling time T_s and zero hold, which is true for the implementation, and using (4.1) together with the solution to the differential equation

$$\dot{\mathbf{x}} = A\mathbf{x}(t) + B\mathbf{u}(t) \quad (4.2)$$

which is given by

$$\mathbf{x}(t) = e^{A(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{A(t-\tau)}B\mathbf{u}(\tau)d\tau, \quad (4.3)$$

the continuous state space model

$$\dot{\mathbf{x}} = A\mathbf{x}(t) + B\mathbf{u}(t) \quad (4.4a)$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}(t) \quad (4.4b)$$

can be written as

$$\mathbf{x}[k+1] = A_d\mathbf{x}[k] + B_d\mathbf{u}[k] \quad (4.5a)$$

$$\mathbf{y}[k] = C_d\mathbf{x}[k] + D_d\mathbf{u}[k] \quad (4.5b)$$

in discrete form [14]. The discrete matrices A_d and B_d are then obtained from the continuous ones according to

$$A_d = e^{AT_s} \quad (4.6)$$

and

$$B_d = \int_0^{T_s} e^{A(T_s-\tau)}Bd\tau, \quad (4.7)$$

respectively, and the C_d and D_d matrices are the same as in the continuous case.

4.2 Linear Quadratic Regulator

A linear quadratic regulator is a state feedback controller within the field of optimal control. The goal with the theory of optimal control is to find a control law that minimises a certain cost function based on a performance criteria [15]. Given a plant on the state space form in (4.5) the time-invariant discrete LQR controller aims at minimising the cost function

$$J = \frac{1}{2} \sum_{k=0}^{\infty} (\mathbf{x}[k]^T Q \mathbf{x}[k] + \mathbf{u}[k]^T R \mathbf{u}[k]) \quad (4.8)$$

where $\mathbf{x}[k]$ is the state vector at time index k , $\mathbf{u}[k]$ is the control input at time index k , Q is a weighting matrix for the states and R is a weighting matrix for the control input [15]. The optimal state feedback control law that minimises the cost function is given by

$$\mathbf{u}[k] = -K\mathbf{x}[k] \quad (4.9)$$

where K is the LQR gain [15]. This optimal gain is found by first solving the discrete algebraic Riccati equation

$$P = A_d^T (P - PB_d(B_d^T PB_d + R)^{-1} B_d^T P) A_d + Q \quad (4.10)$$

to obtain the positive definite matrix P and then selecting the LQR gain as

$$K = (B_d^\top P B_d + R)^{-1} B_d^\top P A_d. \quad (4.11)$$

In order for the control law in (4.9) to be optimal the system has to be stabilisable which means that all uncontrollable states, if there exist any, has to be stable [15]. If no uncontrollable states exist, the system is controllable which implies that it is also stabilisable. For a system on discrete state space form to be controllable the controllability matrix

$$Q_c = \begin{bmatrix} B_d & A_d B_d & A_d^2 B_d & \dots & A_d^{n-1} B_d \end{bmatrix} \quad (4.12)$$

must be full rank, where n is the number of states.

The diagonal elements in the weight matrix R defines the cost for deviation from the states and the diagonal elements in the weight matrix Q defines the cost for the control activity. These elements are the ones to be tuned when tuning the LQR regulator.

In order for the LQR controller to be able to follow a reference $\mathbf{r}[k]$, the control law in (4.9) can be rewritten to

$$\mathbf{u}[k] = -K\mathbf{x}[k] + K_r\mathbf{r}[k] \quad (4.13)$$

where K_r is a reference gain to obtain unit stationary gain from the reference $\mathbf{r}[k]$ to the output $\mathbf{y}[k]$ [16]. The reference gain matrix K_r can be obtained by inserting the control law (4.13) into the state space model in (4.5) and using that $\mathbf{r} = \mathbf{y}$ in steady state when $\mathbf{x}[k+1] = \mathbf{x}[k]$. Then K_r is given by

$$K_r = (C_d(I - A_d + B_d K)^{-1} B_d)^\dagger \quad (4.14)$$

where \dagger denotes pseudoinverse.

4.2.1 LQR for Pressure Feed Reducer

The LQR controller for the software pressure feed reducer is based on the state space model presented in Section 3.4, discretised according to the method described in Section 4.1. Since the objective of the controller is to make sure that the pressure is below a certain limit by moving the spool to its centre position and only limiting the output flow, no overshoot is allowed by the controller. If overshoot in the pressure occurs, the controller cannot remove this since the pressure cannot be lowered by the pressure feed reducer function. When choosing the weights Q and R for the LQR-controller for this application it is therefore important to choose the weights to get a closed loop system with damped behaviour without overshoot. At the same time, it has to be fast enough to limit the pressure before it increases above the limit.

When tuning the LQR-controller it is the relation between the values in Q and the values in R that matters, not just the values. This means that increasing R will give rise to the same behaviour as decreasing all values in Q . Thus, to make the tuning easier, the weight R is kept constant at the value 1 and only the values in Q

are tuned. Decreasing the cost on the pressure state in Q will make the controller penalise high control signals more and deviations from the reference target less. Therefore, only a low force will be added to the end of the valve spool which will result in the software pressure feed reducer being too careful and most of the time the user will not be able to control the valve since the output from the controller will mostly be lower than the user signal. If the pressure state cost instead is increased, higher control signals will be allowed and deviations from the reference target are penalised more. This will make the operator able to control the valve more but the software pressure feed reducer will be less careful.

Tuning of the cost on the pressure velocity in Q will affect how aggressive the controller will be. Increasing the cost on the pressure velocity means that the controller will be slower, and the valve spool will be closed smoother since larger changes in the pressure are penalised harder. If this cost instead is decreased, the controller will be more aggressive and the spool will be closed faster. This also means that the controller will start to close the spool when the measured pressure is closer to the maximum pressure than with a higher cost. The conclusion from this is that the cost on the pressure velocity is important when deciding the behaviour of the pressure feed reducer and how the user will experience the behaviour of the mobile machine. It is also important to tune the cost on the pressure velocity correctly to avoid the pressure exceeding the maximum limit.

4.3 LQR with Integral Action

One problem with the LQR controller presented in Section 4.2 is that it does not guarantee zero reference tracking error in steady state if disturbances are present, or model errors are significant. In order to guarantee zero reference tracking error in steady state if a constant disturbance is present, an integral state \mathbf{x}_i can be added that is the integral of the reference tracking error [14]. This state can be added to (4.8) resulting in the cost-function

$$J_e = \frac{1}{2} \sum_{k=0}^{\infty} \left(\mathbf{x}[k]^T Q \mathbf{x}[k] + \mathbf{x}_i[k]^T Q_i \mathbf{x}_i[k] + \mathbf{u}[k]^T R \mathbf{u}[k] \right) \quad (4.15)$$

where Q_i is the weight for the integral state. The controller that minimises this cost function is a state feedback control law on the form

$$\mathbf{u}[k] = -K \mathbf{x}[k] - K_i \mathbf{x}_i[k] \quad (4.16)$$

where K_i is the integral gain [14]. By combining the state feedback gain K and the integral gain K_i into an extended feedback gain K_e the control-law can be rewritten as

$$\mathbf{u}[k] = -K \mathbf{x}[k] - K_i \mathbf{x}_i[k] = - \underbrace{\begin{bmatrix} K & K_i \end{bmatrix}}_{K_e} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_i \end{bmatrix}. \quad (4.17)$$

The optimal extended feedback gain can be found by designing an LQR-controller for the extended system where the integral state is included in the model. In discrete time the extended model can be written as

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{x}_i[k+1] \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & 0 \\ C_d & I \end{bmatrix}}_{A_e} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_i[k] \end{bmatrix} + \underbrace{\begin{bmatrix} B_d \\ 0 \end{bmatrix}}_{B_e} \mathbf{u}[k] + \begin{bmatrix} 0 \\ -I \end{bmatrix} \mathbf{r}[k] \quad (4.18a)$$

$$\mathbf{y}[k] = \underbrace{\begin{bmatrix} C_d & 0 \end{bmatrix}}_{C_e} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_i[k] \end{bmatrix} + \underbrace{D_d}_{D_e} \mathbf{u}[k]. \quad (4.18b)$$

The extended feedback gain can then be found for the extended model utilising the approach presented in Section 4.2 by using the extended model A_e and B_e and the extended state weight matrix

$$Q_e = \begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}. \quad (4.19)$$

While the approach presented above yields an optimal controller, the cost function (4.15) includes additional states that represent the integral of the tracking errors. These states have a theoretical meaning, but it can be unintuitive to tune the weight on these states since they do not directly represent a physical state of the system. Therefore an alternative to solving the optimal control problem with the cost function (4.15) is to use a standard LQR controller with feedback gain K and then manually tune the integral gain K_i . While this approach no longer is optimal, since integral action is added without being considered in the optimisation problem, it has the advantage that it can be easier to tune in some cases due to the fact that the cost function for the LQR contains states with clear physical meaning. It can also be combined with the reference tracking control law in (4.13) in order to not have to rely solely on the integrator for reference tracking, resulting in the control law

$$\mathbf{u}[k] = -K\mathbf{x}[k] - K_i\mathbf{x}_i[k] + K_r\mathbf{r}[k]. \quad (4.20)$$

If the obtained control law is found to be stabilising, this control law guarantees zero offset reference tracking in steady state in the presence of a constant disturbance [14].

One problem that needs to be addressed when implementing controllers with integral action is integral windup, which is a phenomenon that appears when the control input is saturated, i.e. it is at the maximum or minimum level that can be applied to the plant [17]. Without any modification, the integrator in the controller will keep calculating up or down past the saturation, and it will have to count back before the control signal becomes unsaturated which can lead to bad transient behaviour [17].

There are different ways to solve this problem, called anti-windup techniques. One such technique is back-calculation which aims at recomputing the integral when the control signal is saturated such that it matches the saturation [17]. This recomputation is not done instantaneously, but rather with a time constant of $1/K_b$, and is

achieved by feeding back the difference of the control signal before and after the saturation to the integrator with a gain of K_b . When the control signal is not saturated this feedback is zero, and therefore it does not affect the controller when it is not saturated. However, if the control signal is saturated this feedback is nonzero and pushes the integral towards the value where the control input is at the saturation level.

A block scheme of the LQR controller with integral action and anti-windup can be seen in Figure 4.1. This controller will hereafter be denoted LQRI.

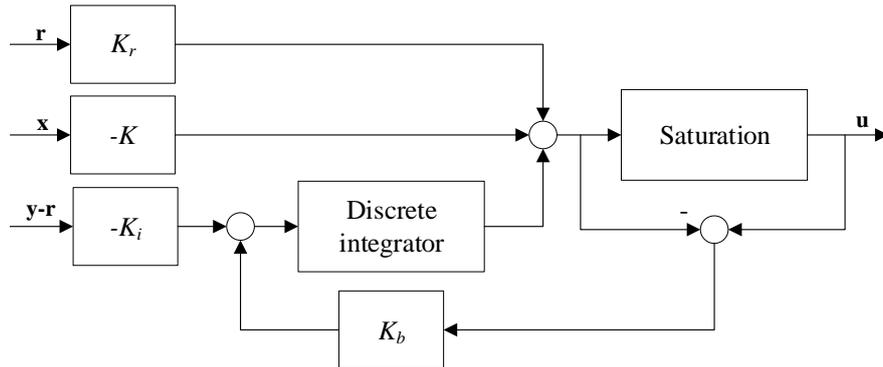


Figure 4.1: Block diagram of an LQR controller with integral action and back-calculating anti-windup.

4.3.1 LQRI for Pressure Feed Reducer

When using the LQR controller with integral action for the software pressure feed reducer the same LQR-gain K and reference tracking gain K_r as for the pure LQR controller is used. This is done since the LQR controller can then be tuned before adding the integrator, and then the integral gain K_i can be tuned such that it gives fast enough disturbance rejection without introducing overshoot or instability. A higher K_i will give a faster disturbance rejection but can introduce overshoot or instability.

In this application the control signal is saturated between 0 and the spool force the operator wants to apply. Therefore, anti-windup is important to apply, and for this the back-calculation technique is used. In order to choose the anti-windup gain K_b it can be observed that the LQRI controller is essentially a PID controller with an added term for the pressure acceleration. This is because the states are the pressure, pressure derivative and pressure acceleration, and the integrator is integrating the pressure offset from the maximum pressure. It has been observed that the LQR gain for the pressure acceleration is significantly smaller than the ones for the pressure and pressure derivative. Therefore, the LQRI controller is approximately a PID controller. A rule of thumb for choosing K_b for a PID controller is to choose

$$K_b = \frac{1}{\sqrt{T_i T_d}} \quad (4.21)$$

where T_i is the time constant for the integration part and T_d is the time constant for the derivative part of the PID controller [17]. The time constants correspond to the LQR gains as

$$T_i = \frac{K(1)}{K_i} \quad (4.22a)$$

$$T_d = \frac{K(2)}{K(1)} \quad (4.22b)$$

where $K(1)$ and $K(2)$ are the first and second entries in the LQR-gain respectively. This results in the anti-windup gain

$$K_b = \frac{1}{\sqrt{T_i T_d}} = \sqrt{\frac{K_i}{K(2)}}. \quad (4.23)$$

In addition to the anti-windup, the LQRI controller is equipped with a feature that disables the integrator completely when the LQR-part of the controller wants to apply a control signal u_{LQR} that is larger than the operator signal for a certain amount of time T_{ri} . When this happens the integrator is set to 0. This is done in order to start the controller as soon as the LQR-part tells it to override the user, without having to wait for the anti-windup. In this way the integrator in the controller is mainly responsible for disturbance rejection, and does not effect when the controller is overriding the user. The time T_{ri} is included to not immediately reset the integrator when the LQR-part wants a higher spool force than the operator again. The anti-windup scheme should in that case constrain the integrator instead. When the time T_{ri} has elapsed, the integrator is reset again in order to start the controller properly the next time it needs to.

A block diagram of the LQRI controller for the pressure feed reducer with back-calculating anti-windup and integrator resetting can be seen in Figure 4.2, where u_o is the signal the operator wants to apply and can be calculated as

$$u_o(t) = A_{cs} p_p(i_o(t)) - k_s x_{dba}. \quad (4.24)$$

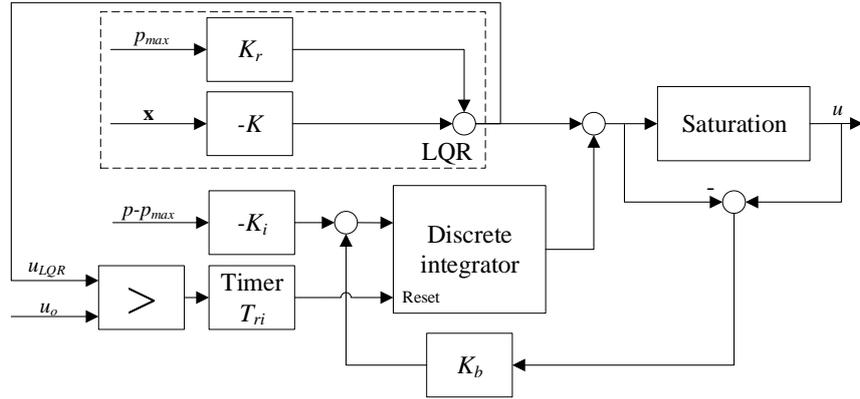


Figure 4.2: Block diagram of the LQRI controller with back-calculating anti-windup for the software pressure feed reducer.

4.4 \mathcal{H}_∞ Controller

The \mathcal{H}_∞ controller is a controller that belongs to the robust branch of control theory. That a control system is robust means that mismatches between the real physical system and the model used for controller design does not affect the performance of the controller significantly [18]. This makes robust control systems suitable for cases when a model of the physical system is available and the characteristics of the model uncertainties are known. In this section the continuous version of \mathcal{H}_∞ controller will be presented and when applied to the software pressure feed reducer the controller will be discretised using the method described in Section 4.1 before being implemented.

In order to perform robust control, the plant and the controller must be formulated in a specific way. The generic design of the control configuration used for controller design can be seen in Figure 4.3, where P is the generalised plant and K is the generalised controller [18]. In the model configuration \mathbf{w} are the exogenous inputs and consists of disturbance \mathbf{d} , noise \mathbf{n} and reference \mathbf{r} according to

$$\mathbf{w} = \begin{bmatrix} \mathbf{d} \\ \mathbf{r} \\ \mathbf{n} \end{bmatrix} \quad (4.25)$$

and \mathbf{z} are the exogenous outputs which are the performance outputs that are important for the system [18]. The system in Figure 4.3 can mathematically be described according to

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{v} \end{bmatrix} = P \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix}. \quad (4.26)$$

The total closed loop system is called N and the goal of the \mathcal{H}_∞ controller is to minimise the closed loop \mathcal{H}_∞ norm from \mathbf{w} to \mathbf{z} , i.e. $\|N(j\omega)\|_\infty$.

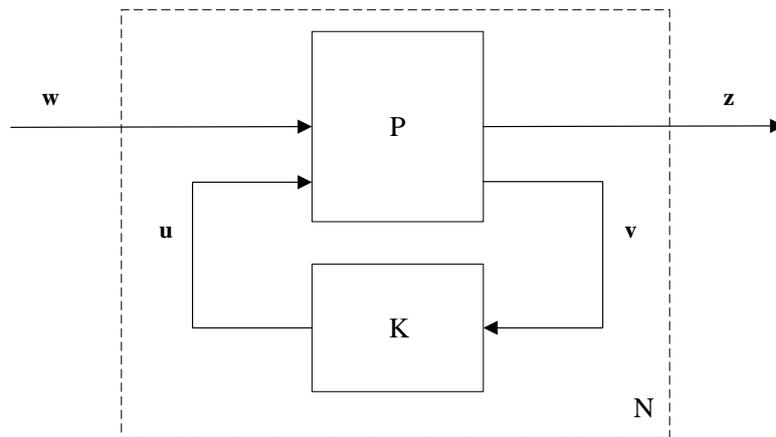


Figure 4.3: Block diagram of a general robust controller system.

The exogenous inputs \mathbf{w} and outputs \mathbf{z} are usually weighted with general weights W_z and W_w respectively that are included in the plant P according to Figure 4.4, where \tilde{P} is the physical plant without the weights. The weights are included in order to have a meaningful controller synthesis problem and bring additional information for the controller design [18]. W_z and W_w are transfer function matrices and are usually frequency dependent and chosen such that the weighted signals \mathbf{w} and \mathbf{z} have a magnitude of 1. The weights inject information from the real world into the controller design and should be selected based on knowledge from the physical system. This makes it possible to describe the disturbance, reference and noise in a more specific way, which is necessary to get an \mathcal{H}_∞ controller.

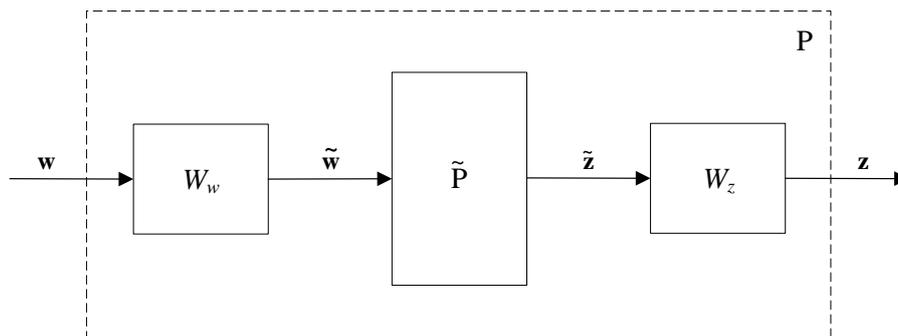


Figure 4.4: Block diagram showing the generic plant P with weights added on inputs and outputs.

The general weight W_w describes the frequency content of the input signals \mathbf{d} , \mathbf{n} and \mathbf{r} and therefore it usually consists of different weights for each signal, i.e. a weight W_d for the disturbance, a weight W_n for the noise and a weight W_i for the reference [18]. The performance output signal \mathbf{z} may also consist of several different signals and therefore the weight W_z also usually consists of different separate

weights. The weights in W_z are used to describe how important different frequencies in the performance output signals are. For example, if a low pass filter is used on a performance output, low frequencies will be penalised more than high frequencies for that output.

As mentioned previously the goal of the \mathcal{H}_∞ controller is to find a controller K such that the closed loop norm $\|N(j\omega)\|_\infty$ is minimised. The transfer function $N(j\omega)$ from \mathbf{w} to \mathbf{z} can be found using the linear fractional transformation $F_l(P, K)$ which is given by

$$F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} = N(j\omega) \quad (4.27)$$

where P_{11} , P_{12} , P_{21} and P_{22} are parts of the general plant P shown in (4.26). The controller K that minimises $\|N(j\omega)\|_\infty$ can be found using algorithms based on two Riccati equations and a state space solution [19],[20]. The optimal controller K obtained from this algorithm is a transfer function and it holds that

$$\|N(j\omega)\|_\infty \leq \gamma \quad (4.28)$$

where γ is an upper bound and performance metric for the \mathcal{H}_∞ controller [18]. To get the optimal solution and lowest value for γ an algorithm called γ -iteration can be used [18].

4.4.1 \mathcal{H}_∞ Controller for Pressure Feed Reducer

To construct an \mathcal{H}_∞ controller for the software pressure feed reducer the plant for the valve and load together with the controller has to be formulated in the general form presented in Figure 4.3 and weights for all signals has to be selected. The complete closed loop system used for the controller-design can be seen in Figure 4.5 where G_v is the plant for the valve model given in Section 3.1 and G_l is the plant for the load model given in Section 3.2.

As can be seen in Figure 4.5 the input u to the valve model is the difference between the normalised input from the operator, $u_{o,norm}$ weighed with W_{uo} and the input signal from the controller, u_k . This differ a bit from how the LQR-controller is working, which directly gives the signal that goes to the valve. This means that when the \mathcal{H}_∞ controller increases its control signal, the signal sent to the valve will decrease, assuming that the operator signal is constant. Representing the input signal to the valve in this way is more logical, since a more active control signal, i.e. a higher spool force, should result in a more closed spool. This change in input signal is done to make it more intuitive.

The weights W_d , W_i and W_n in Figure 4.5 corresponds to the general weight W_w described previously in this section. W_d is added to the disturbance d , which represents flow disturbances, and describes the frequency content in this signal. Since the flow disturbances are of low frequency W_d is chosen as a low pass filter. W_i is the weight on the reference input, i.e. the maximum allowed pressure in this case. It is also chosen as a low pass filter with a gain corresponding to the maximum pressure

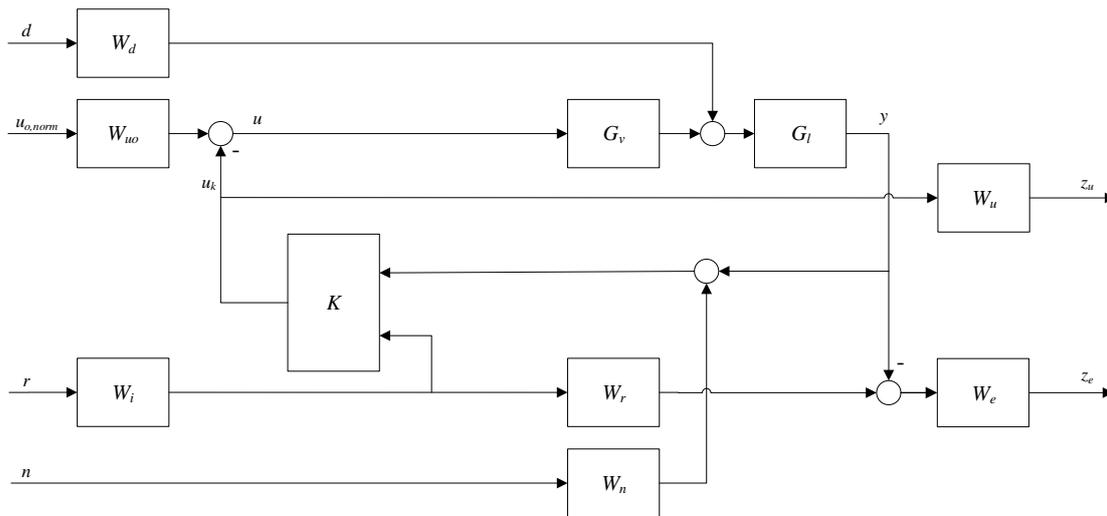


Figure 4.5: Block diagram of closed loop system used for designing a \mathcal{H}_∞ controller for the software pressure feed reducer.

and low crossover frequency since the reference is constant during running. W_n is a weight on the measurement noise on the pressure and represents the content of the measurement noise. Since this noise is assumed to be both low and high frequency W_n is chosen as a constant gain on 1 bar.

W_r in Figure 4.5 is a weight on the weighted reference and is a transfer function describing the desired closed loop behaviour from the weighted reference to the output pressure. For the software pressure feed reducer W_r is chosen as a first-order transfer function with a time constant of 100 ms. W_{uo} is a weight on the signal from the operator and represents the frequency content of this signal which is assumed to be low frequency. Due to this W_{uo} is chosen as a low pass filter.

The general weight W_z described previously in this sections consists of the two weights W_e and W_u for this application and can be seen in Figure 4.5. W_u is added to the control signal from the controller and describes which frequencies of the control signal that are penalised. It is chosen as a high pass filter since high frequency changes in the control signal are more important to penalise than low-frequency changes. W_e is a weight on the reference tracking error and is therefore chosen as a low pass filter.

Once all weights were designed, the generalised plant P was built in MATLAB using the command `sysic` and the controller K was computed using the command `hinfsyn`.

4.5 \mathcal{H}_∞ Loop-Shaping

Another control algorithm similar to the standard \mathcal{H}_∞ controller is the robust \mathcal{H}_∞ Loop-Shaping controller. This controller is a combination of the \mathcal{H}_∞ robust stabilisation controller described in Section 4.4 and loop shaping [18]. The general idea is to augment the open loop plant to control, in order to obtain a desired shape of the open loop frequency response and then robustly stabilise the augmented plant using \mathcal{H}_∞ optimisation [18]. In this section the continuous version of the \mathcal{H}_∞ Loop-Shaping controller will be presented and when applied to the software pressure feed reducer it will be discretised using the method described in Section 4.1 before being implemented.

The goal of the two degrees of freedom \mathcal{H}_∞ Loop-Shaping controller is to design a controller on the form

$$K = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \quad (4.29)$$

and set up the final closed loop system as depicted in Figure 4.6 where W_1 is a pre-compensator weight for the plant $G(s)$. This weight should be chosen such that a desired open loop frequency response is obtained [18]. Usually this means a slope of -1 at the desired frequencies in the Bode plot and high gain at low frequencies together with steeper slope at high frequencies [18]. W_i in Figure 4.6 is a weight to make the prefilter $K_1 W_i$ give the closed loop system exact model matching at steady state [18].

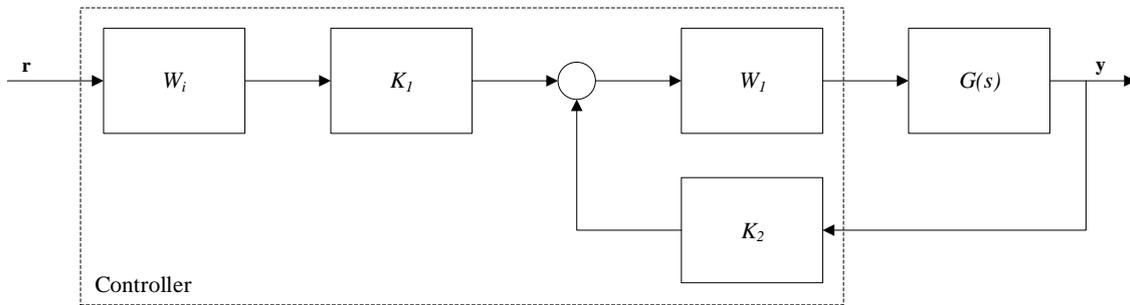


Figure 4.6: General block diagram of closed loop system for two degrees of freedom \mathcal{H}_∞ Loop-Shaping controller.

The controller $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$ can be synthesised using the closed loop illustrated in Figure 4.7 [18]. In this figure the augmented plant

$$G_s(s) = G(s)W_1 \quad (4.30)$$

is represented as a plant with a normalised left coprime factorisation that is perturbed and denoted $G_p(s)$. That the plant $G_s(s)$ is a normalised left coprime factorisation means that it can be written on the form

$$G_s(s) = M_s(s)^{-1}N_s(s) \quad (4.31)$$

and that

$$M_s M_s^* + N_s N_s^* = I \quad (4.32)$$

holds [18]. The perturbed plant $G_p(s)$ is then given by

$$G_p(s) = (M_s + \Delta_{M_s})^{-1}(N_s + \Delta_{N_s}) \quad (4.33)$$

where Δ_{M_s} and Δ_{N_s} are unknown and stable transfer functions representing uncertainty in the plant $G_s(s)$.

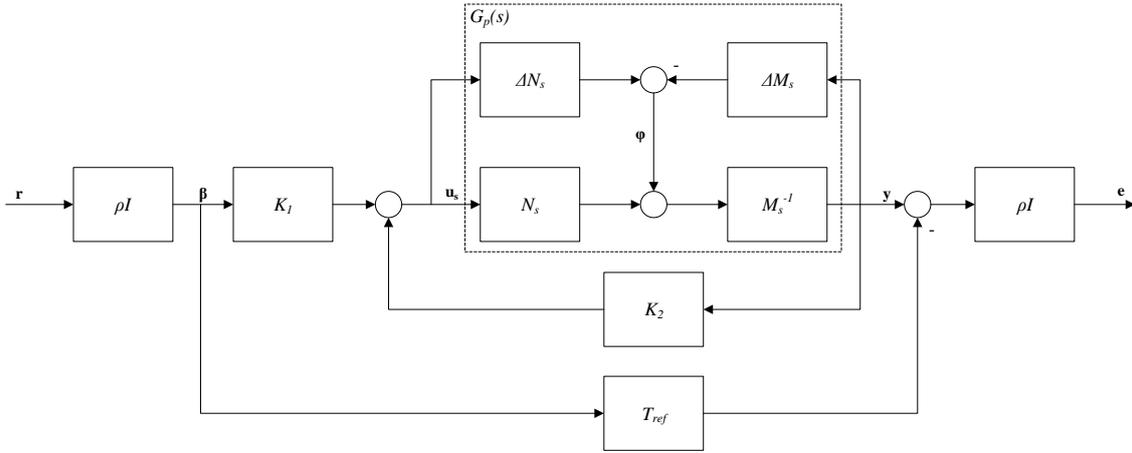


Figure 4.7: Block diagram of closed loop system used for designing two degrees of freedom \mathcal{H}_∞ Loop-Shaping controller.

The design objective for the closed loop system in Figure 4.7 is then to find a controller $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$ minimising the \mathcal{H}_∞ norm of the transfer function from the exogenous inputs $\begin{bmatrix} \mathbf{r}^\top & \boldsymbol{\varphi}^\top \end{bmatrix}$ to the exogenous outputs $\begin{bmatrix} \mathbf{u}_s^\top & \mathbf{y}^\top & \mathbf{e}^\top \end{bmatrix}$ in a similar way as in Section 4.4 [18]. The control signal \mathbf{u}_s to the augmented plant is given by

$$\mathbf{u}_s = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{y} \end{bmatrix} \quad (4.34)$$

where $\boldsymbol{\beta}$ is the reference \mathbf{r} scaled with ρ and \mathbf{y} is the measured output. K_2 is the feedback controller and K_1 is a prefilter with the purpose to ensure that

$$\|(I - G_s(s)K_2)^{-1}G_s(s)K_1 - T_{ref}\|_\infty \leq \gamma\rho^{-2}, \quad (4.35)$$

where ρ is a scalar tuning parameter and γ is the same upper bound as in Section 4.4. A higher ρ means that more weight is added to model matching instead of robustness during the optimisation [21]. T_{ref} is a reference transfer function for the closed loop and should represent the desired closed loop behaviour in the time domain and is specified during the design phase [21].

The controller design problem can now be formulated on the general form presented in (4.26) according to

$$\begin{bmatrix} \mathbf{u}_s \\ \mathbf{y} \\ \mathbf{e} \\ \beta \\ \mathbf{y} \end{bmatrix} = \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}}_P \begin{bmatrix} \mathbf{r} \\ \varphi \\ \mathbf{u}_s \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & I \\ 0 & M_s^{-1} & G_s(s) \\ -\rho^2 T_{ref} & \rho M_s^{-1} & \rho G_s(s) \\ \rho I & 0 & 0 \\ 0 & M_s^{-1} & G_s(s) \end{bmatrix}}_P \begin{bmatrix} \mathbf{r} \\ \varphi \\ \mathbf{u}_s \end{bmatrix} \quad (4.36)$$

where P is obtained from the block diagram in Figure 4.7 [18]. The generalised plant P can then be used in standard \mathcal{H}_∞ algorithms, like the one described in Section 4.4, to obtain the controller gain $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$ in Figure 4.6 [22].

Once the controller gain K is obtained, the prefilter weight W_i in Figure 4.6 has to be calculated in order to have a complete \mathcal{H}_∞ Loop-Shaping controller. As previously mentioned W_i is a weight that makes the prefilter $K_1 W_i$ give the closed loop system exact model matching with the reference model T_{ref} at steady state [18]. Thus, the weight can be calculated according to

$$W_i = \left[(I - G_s(s)K_2(s))^{-1} G_s(s)K_1(s) \right]^{-1} T_{ref}(s)|_{s=0} \quad (4.37)$$

and the complete controller in Figure 4.6 is obtained.

4.5.1 \mathcal{H}_∞ Loop Shaping for Pressure Feed Reducer

When designing an \mathcal{H}_∞ Loop-Shaping controller for the software pressure feed reducer the reference model T_{ref} , the pre-compensator weight W_1 and the scalar tuning parameter ρ has to be chosen. T_{ref} is chosen as a second order transfer function with a steady state gain of 1, undamped natural frequency of 5 Hz and damping factor of 1 according to

$$T_{ref}(s) = \frac{(5 \cdot 2\pi)^2}{s^2 + 2 \cdot 5 \cdot 2\pi s + (5 \cdot 2\pi)^2} \quad (4.38)$$

since this gives a desired behaviour for the closed loop system. This is a desired behaviour since a damping factor of 1 represents no overshoot and a natural frequency of 5 Hz represents a reasonable response time for the pressure feed reducer.

The weight W_1 is chosen as a static gain with different values for workport A and workport B since this results in a open loop frequency response that is common when using \mathcal{H}_∞ Loop-Shaping and resulted in a good behaviour during simulation. A Bode plot of the open loop frequency response corresponding to workport A can be seen in Figure 4.8. As can be seen the gain is high for low frequencies and the slope is higher for higher frequencies. This is desired as mentioned earlier in this section.

The value 1 for the scalar parameter ρ turned out to work well and therefore that was used.

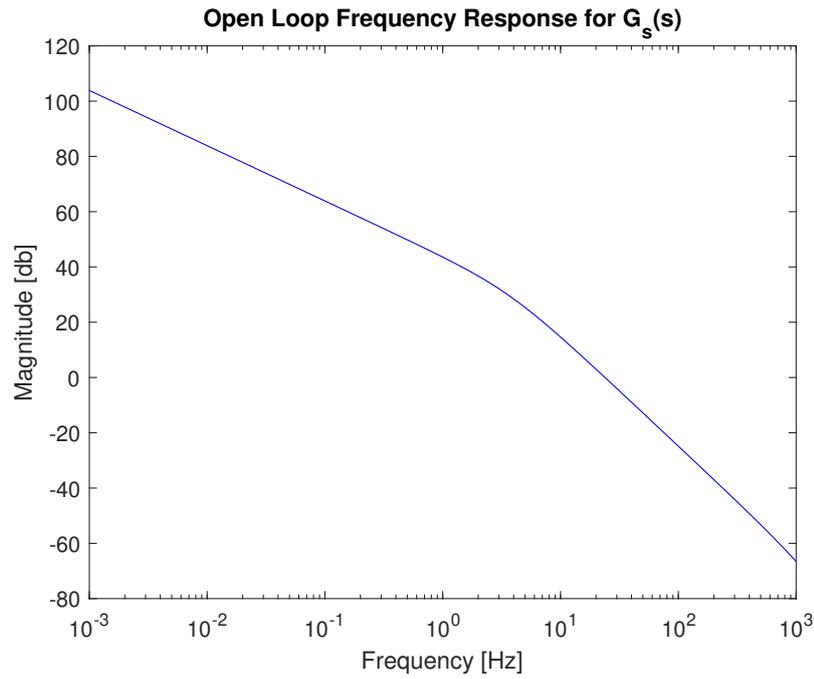


Figure 4.8: Bode plot of the open loop frequency response for $G_s(s)$ corresponding to workport A.

After deciding the three parameters for the \mathcal{H}_∞ Loop-Shaping controller the controller K was calculated in MATLAB using the command `hinfsyn` just like in Section 4.5, and then discretised using zero order hold as presented in Section 4.1.

4.6 Model Predictive Controller

Model predictive control is an optimal control scheme where a cost function is minimised while obeying constraints, such as state and input constraints [23]. It achieves this by optimising the control signal over a receding horizon. This means that, at every time step, the controller predicts the system response for a finite prediction horizon of N time steps into the future, depending on future control inputs [23]. Based on the prediction, cost function and constraints the controller finds the optimal sequence of control inputs over the prediction horizon that obeys the constraints. The first element of the optimal control sequence found is then applied to the process, and the rest of the sequence is discarded. This process is then repeated at every time step.

A special case of MPC is the so called linear quadratic MPC. In linear quadratic MPC, the system model is linear and discrete as in (4.5), the cost function is quadratic and the constraints are linear [23]. If the constraints on the control signal and the states are upper and lower bounds the optimisation problem that is solved at each time step k can be expressed as

$$\min_{\substack{\mathbf{u}[k:k+N-1], \\ \mathbf{x}[k+1:k+N]}} \left(\mathbf{x}[k+N]^\top Q_f \mathbf{x}[k+N] + \sum_{i=k}^{k+N-1} \left(\mathbf{x}[i]^\top Q \mathbf{x}[i] + \mathbf{u}[i]^\top R \mathbf{u}[i] \right) \right) \quad (4.39)$$

subject to the constraints

$$\mathbf{x}[k+i+1] = A_d \mathbf{x}[k+i] + B_d \mathbf{u}[k+i], \quad \forall i \in [0, N-1] \quad (4.40a)$$

$$\mathbf{u}_{min} \leq \mathbf{u}[k+i] \leq \mathbf{u}_{max}, \quad \forall i \in [0, N-1] \quad (4.40b)$$

$$\mathbf{x}_{min} \leq \mathbf{x}[k+i] \leq \mathbf{x}_{max}, \quad \forall i \in [1, N] \quad (4.40c)$$

where Q_f is the weighting matrix for the state at the end of the prediction horizon, Q is the weighting matrix for the state during the prediction horizon and R is the weighting matrix for the control input. \mathbf{u}_{min} and \mathbf{u}_{max} are the lower and upper bounds for the control signal, and \mathbf{x}_{min} and \mathbf{x}_{max} are the lower and upper bounds for the states.

Since the cost function (4.39) is quadratic, and the constraints (4.40) are linear the optimisation problem is a so called quadratic program [23]. This is desired since efficient algorithms exist for solving such optimisation problems.

One potential problem with MPC is that the optimisation problem can become infeasible for some state combinations due to the constraints, meaning that it is not possible to find a sequence of control inputs that obey both the input and state constraints. One way to avoid infeasibility is to make the state constraints into soft constraints [23]. This can be done by adding a so called slack variable ϵ that represents how much the worst constraint is violated [24]. This then changes the

optimisation problem to

$$\min_{\substack{\mathbf{u}[k:k+N-1], \\ \mathbf{x}[k+1:k+N], \\ \epsilon}} \left(\mathbf{x}[k+N]^\top Q_f \mathbf{x}[k+N] + \sum_{i=k}^{k+N-1} \left(\mathbf{x}[i]^\top Q \mathbf{x}[i] + \mathbf{u}[i]^\top R \mathbf{u}[i] \right) + \rho_\epsilon \epsilon^2 \right) \quad (4.41)$$

subject to the constraints

$$\mathbf{x}[k+i+1] = A_b \mathbf{x}[k+i] + B_d \mathbf{u}[k+i], \quad \forall i \in [0, N-1] \quad (4.42a)$$

$$\mathbf{u}_{min} \leq \mathbf{u}[k+i] \leq \mathbf{u}_{max}, \quad \forall i \in [0, N-1] \quad (4.42b)$$

$$\mathbf{x}_{min} - \epsilon \mathbf{V}_{min}^x[i] \leq \mathbf{x}[k+i] \leq \mathbf{x}_{max} + \epsilon \mathbf{V}_{max}^x[i], \quad \forall i \in [1, N] \quad (4.42c)$$

where ρ_ϵ is the weight for the slack variable and $\mathbf{V}_{min}^x[i]$ and $\mathbf{V}_{max}^x[i]$ are parameters that determine how soft the constraints are at i time steps into the future [24]. A higher ρ_ϵ makes the constraints harder, i.e. it penalises the controller more for overriding the constraints. \mathbf{V}_{min}^x and \mathbf{V}_{max}^x can be used to make the constraints harder or softer at different prediction lengths. A higher value makes the constraints softer at the specific prediction length.

4.6.1 MPC for Pressure Feed Reducer

The main advantage with the MPC controller over the LQR and \mathcal{H}_∞ controllers is that it takes constraints on control signals and states into account. This makes it suitable for the software pressure feed reducer since there are constraints present in this application, namely the following:

- The control signal from the controller is not allowed to be higher than the operator input.
- The control signal from the controller is not allowed to be negative, since the controller is not allowed to drive the function backwards.
- The pressure in the workport is not allowed to be higher than a constant maximum value.

The implemented MPC controller is a linear quadratic MPC based on the model in Section 3.4 and discretised according to the method described in Section 4.1. It was implemented in MATLAB and Simulink using the Model Predictive Control Toolbox from MathWorks.

Since the controller is not supposed to drive the pressure in the workport to the maximum value, but rather just keep it below a set value, no cost on the states is used in the controller, i.e. $Q = Q_f = 0$. Furthermore, the control input from the controller, i.e. the force on the spool, is desired to be close to the operator input. Therefore, the difference $u[k] - u_o[k]$ is penalised in the cost function instead of the control input directly, where

$$u_o[k] = A_{cs} p_p(i_o[k]) - k_s x_{dA} \quad (4.43)$$

is the operator input recalculated as a force acting on the spool. In order to do this, it is assumed that the user input is kept constant over the whole prediction horizon since no knowledge of future inputs from the user is present.

The constraints on the controller input are hard constraints, since it is not allowed to give a negative control signal, and the control signal is not allowed to go above the user input. Here it is also assumed that the user input is kept constant over the whole prediction horizon.

As suggested previously, in order to avoid infeasibility problems, the state constraints are converted into soft constraints. There is also only one state constraint, namely that the pressure in the workport is not allowed to be higher than a constant maximum value. This results in the optimisation problem

$$\min_{\substack{u[k:k+N-1], \\ \mathbf{x}[k+1:k+N], \\ \epsilon}} \left(\sum_{i=k}^{k+N-1} \left(R(u[i] - u_o[k])^2 \right) + \rho_\epsilon \epsilon^2 \right) \quad (4.44)$$

subject to the constraints

$$\mathbf{x}[k+i+1] = A_b \mathbf{x}[k+i] + B_d u[k+i], \quad \forall i \in [0, N-1] \quad (4.45a)$$

$$0 \leq u[k+i] \leq u_o[k], \quad \forall i \in [0, N-1] \quad (4.45b)$$

$$p_A[k+i] \leq p_{max}[k] + \epsilon V_{max}^p[i], \quad \forall i \in [1, N] \quad (4.45c)$$

that is solved at every time step in a receding horizon fashion.

This MPC controller has some tuning parameters, namely the prediction horizon N , input weight R , the slack variable weight ρ_ϵ and the constraint softness V_{max}^p for the pressure constraints. The prediction horizon N determines how far in the future the controller is looking. Making it larger can improve the performance of the controller, but it also increases the computational complexity of the optimisation problem. Thus N should be chosen large enough to be able to predict the behaviour of the model far enough, while not too large to make the computational complexity too high.

Making the input weight R larger will make the controller penalise deviations from the operator input more, and thus it will make the controller allow the operator control the valve more. Thus, it will make the controller wait longer until it closes the valve when the pressure increases and close it faster. The slack variable weighting ρ_ϵ has the opposite effect since it penalises violations of the pressure constraints, and is therefore kept constant since the same behaviour can be obtained by changing R .

Finally, the constraint softness V_{max}^p can be tuned differently for different long predictions, which is useful since it can make the constraints softer for longer predictions. This can be appropriate when the model is not very accurate for long predictions since violations of the constraints further into the future will not be penalised as much as violations of the constraints closer in time.

One problem with MPC is that it has a high computational complexity since it has to solve a quadratic constrained optimisation problem, such as the one presented in (4.44) and (4.45), at every time step. In the case of the software pressure feed reducer, the sampling time is 10 ms so the controller must be able to solve the opti-

misation problem within 10 ms. In order to guarantee that the optimisation problem can be solved within the sampling time, the optimisation solver can be stopped after a fixed number of iterations, K_{max} , and the solution obtained after these iterations can be used for control [25]. This solution will be a so called suboptimal solution since the optimisation solver has not done enough iterations to find the optimal solution. Experiments have, however, shown that such a suboptimal solution still gives high quality control, even with K_{max} as low as 3 to 10 iterations [25]. Furthermore, simulations of the software pressure feed reducer using the MPC controller with $K_{max} = 10$ iterations gave no noticeable change in performance compared to using the optimal solution to the optimisation problem.

4.7 PID Controller with Bumpless Transfer

A PID controller in continuous time has the form

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t) \quad (4.46)$$

where K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain and

$$e(t) = r(t) - y(t) \quad (4.47)$$

is the reference tracking error [16]. By approximating the integral as a sum, and using the definition for the derivative, the PID controller can be discretised as

$$u[k] = K_p e[k] + I[k] + K_d \frac{e[k] - e[k-1]}{T_s} \quad (4.48a)$$

$$e[k] = r[k] - y[k] \quad (4.48b)$$

$$I[k] = I[k-1] + K_i T_s e[k] \quad (4.48c)$$

where $I[k]$ is the approximation of $K_i \int_0^t e(\tau) d\tau$.

In some applications an operator can manually change the control signal and then enable the PID controller. In order to not create a bump in the control signal when switching from the operator control signal $u_o[k]$ to the PID control signal $u[k]$ a bumpless transfer feature is needed. In some applications an appropriate way to achieve this is to reset the integrator to zero when the operator is in control, and calculate K_p such that the output from the controller is the same as the operator control signal when the controller is first enabled. This K_p can be found by setting $u[k] = u_o[k]$ in (4.48a) and solving for K_p which yields

$$K_p = \frac{u_o[k] - K_d(e[k] - e[k-1])/T_s}{e[k]} \quad (4.49)$$

where $I[k]$ was set to zero since the integral is reset when the operator is in control. This K_p is then used until the operator takes control of the control signal again. Hereafter when referring to a PID controller, this bumpless transfer feature is included in the controller. The PID controller with bumpless transfer will also be referred to as bumpless PID.

In PID controllers integral windup can be a problem when the control signal is saturated. This was discussed in Section 4.3 and the back-calculation technique can be applied for the PID controller as well, with the gain from the rule of thumb in (4.21). For the PID controller described here the time constants are

$$T_i = \frac{K_p}{K_i} \tag{4.50a}$$

$$T_d = \frac{K_d}{K_p} \tag{4.50b}$$

and therefore the anti-windup gain can be calculated as

$$K_b = \frac{1}{\sqrt{T_i T_d}} = \sqrt{\frac{K_i}{K_d}}. \tag{4.51}$$

A block diagram of the final PID controller with back-calculating anti-windup can be seen in Figure 4.9.

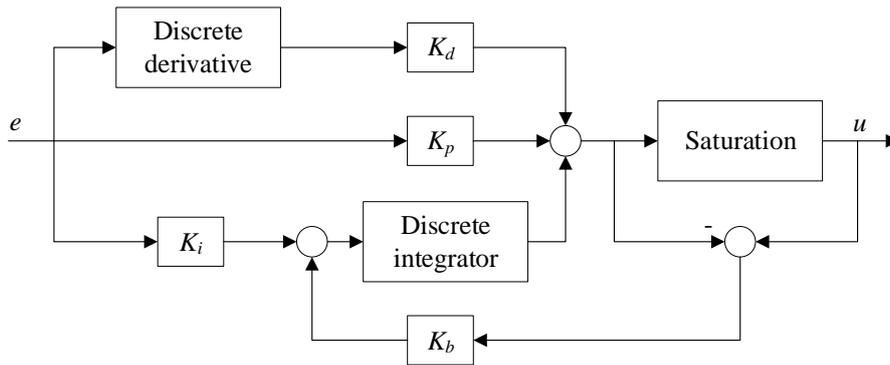


Figure 4.9: Block diagram for PID controller with back-calculation anti-windup.

4.7.1 PID Controller for Pressure Feed Reducer

In order to apply the PID controller with bumpless transfer to the software pressure feed reducer a way of determining if the controller or the operator should control the valve is needed. One way to do this is to predict what the pressure will be a certain amount of time T_{pred} in the future using the model, given that the input to the valve is constant. If the predicted pressure is above the maximum allowed pressure, the controller is enabled, and otherwise it is disabled. This will ensure that the controller only takes control over the valve if the pressure is predicted to be above the maximum pressure.

A simpler approach would be to enable the controller if the pressure gets within a certain range of the maximum pressure. However, the prediction approach has the advantage that it takes both the user input and the current state into account. This makes it take control over the valve at different pressure levels depending on the situation.

Predicting the pressure can be done by iteratively applying the discretised version of the state space model in (3.18) to the state T_{pred}/T_s times, starting from the current state. The current operator signal is used as input throughout the entire prediction.

The prediction time T_{pred} can be seen as a tuning parameter. A higher value means that the prediction looks further into the future, and therefore the controller will be enabled earlier, meaning that it will take control over the valve at a lower pressure level. Therefore T_{pred} should be set large enough for the controller to have time to close the valve before the pressure exceeds the maximum pressure. It should also be small enough such that the controller does not override the user unnecessarily early.

Combining the prediction approach with the bumpless transfer method of adjusting K_p may work well. It makes the controller use a higher K_p if it is enabled close to the pressure limit when the operator signal is high, and a lower K_p if it is enabled far from the pressure limit when the operator signal is low. This is intuitively desirable since it will make the controller close the valve faster if it is enabled close to the limit, and slower if it is far from it.

4.8 Controller Output Conversion

The output from the controllers developed in this chapter must be processed before it can be sent to the pilot valve controlling the movement of the valve spool. A block diagram of this process can be seen in Figure 4.10 where the user input $i_o(t)$ and the spool force $u(t)$ from the controller is seen as inputs and the input current $i(t)$ to the pilot valve is seen as the output of this process.

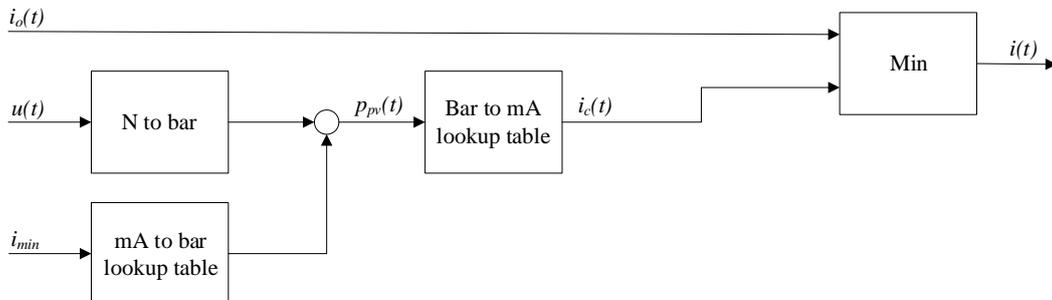


Figure 4.10: Block diagram showing how the spool force $u(t)$ from a controller is used together with the input current, $i_o(t)$, from the operator.

The first step is to convert the spool force into the correct unit. The pilot valve used for implementation is controlled with a voltage of 24 V and therefore expects a signal in current given in milliamperes in the range between 0 and 650 mA [12]. This means that the spool force obtained from the controller has to be converted first into pilot pressure using the cross section area of the spool. Then the pilot pressure to overcome the spool deadband mentioned in Chapter 3 is added. This is not added directly as a pressure, but rather a pressure calculated from a set current i_{min} using the lookup table for the pilot valve. The reason for this is that the current

required is not exactly the same for all valves due to manufacturing differences, and the current i_{min} is a parameter that is available in the original software controlling the valve. This parameter is tuned for every valve, and therefore using it instead of a fixed pilot pressure will make the controllers use the correct offset to overcome the deadband for every specific valve. This results in the pilot pressure $p_{pv}(t)$ in bar desired from the controller as

$$p_{pv}(t) = \frac{u(t)}{A_{cs}} \cdot 10^{-5} + p_p(i_{min}) \cdot 10^{-5} \quad (4.52)$$

where the factor 10^{-5} is used to convert the pilot pressure from pascal to bar. The pilot pressure is then converted into a current $i_c(t)$ using the inverse of the pilot valve lookup table.

After converting the control signal to a pilot valve current it is compared to the current requested from the operator using a minimum-function before being applied to the valve. This is done since the aim is to limit the output pressure from the valve, not to actually drive it to the set maximum pressure. It is also a safety feature since it does not allow the controller to open the valve more than the operator desires, which eliminates the risk for the controlled function to move faster than the operator wants.

5

Simulation of Controllers

In this chapter simulation results for the different controllers described in Chapter 4 are presented. First the model that has been used for simulation is introduced and then the results from the simulations are presented using graphs and different performance metrics. Only results from simulations made on workport A of the valve are presented since the behaviour in workport B is similar to port A in simulation. Besides this only results from one maximum pressure level are shown since the behaviour is similar for different pressure levels.

All simulations have been made in Simulink.

5.1 Model Used for Simulation

The model used for simulation is based on the mathematical model given in (3.13) where the area curve for the spool opening $A_A(x_s(t))$ is obtained from Parker Hannifin. The area curve corresponds to the spool mounted in the K170LS valve used for implementation which makes the simulation model more similar to the physical valve.

The input signal to the simulation model is a force in newton representing the force added to the spool in the physical K170LS valve and the operator signal is a current in milliamperes. Since the model should be simulated together with the operator input the controller signal first has to be converted to pilot valve current and compared to the operator input as in Section 4.8. Then the pilot valve current has to be converted back to spool force in order to be applied to the model in (3.13). This can be done using a mapping from current to pilot pressure together with a conversion from pilot pressure to spool force. This mapping is obtained from Parker Hannifin and is based on data from the type of pilot valve that is used in the K170LS valve and makes the simulations more similar to the physical valve.

As mentioned in Section 3.2 the load model used in the complete model given in (3.13) does not exactly represent the physical system in all cases. The load is assumed to be a fixed volume connected to the valve which is true in many of the cases when the software pressure feed reducer should limit the pressure, but not in all cases. Sometimes the maximum pressure might be high enough to make the cylinder piston in the cylinder attached to the valve start moving slowly before the maximum pressure is reached and, thus, moving the boom on the backhoe. This means that the flow out of the valve will increase when the valve spool opening is increased, but it is

not necessary that the pressure starts to increase due to the cylinder piston starting to move. The model that the developed control algorithms are based on does not take this into account and in order to examine how they handle this model error, a disturbance is added to the pressure derivative state in the simulation model. Adding a negative disturbance to the pressure derivative state gives the same result as adding a disturbance to the flow, which can be seen from (3.11). The disturbance is added in form of a constant offset which represents a movement of the cylinder piston with a constant velocity. To obtain a reasonable value for the disturbance, half the maximum velocity of the cylinder piston on the backhoe used for implementation can be used together with the corresponding cylinder dimensions found in the manual belonging to the backhoe [7]. The disturbance $d_{\dot{p}_A}$ in bar/s can then be calculated according to

$$d_{\dot{p}_A} = -\frac{1}{2} \cdot \frac{\beta}{V_1} \cdot \frac{V_{cyl}}{T_{stroke}} \cdot 10^{-5} \approx -157 \text{ bar/s} \quad (5.1)$$

where V_{cyl} is the chamber volume of the boom cylinder on the backhoe and T_{stroke} is the minimum stroke time of the boom on the backhoe.

5.2 Simulation Results

When simulating the controllers developed in Chapter 4 a constant operator signal on 500 mA have been used both in the case without disturbance and in the case with disturbance. In the beginning of each simulation the valve is closed and the pressure in the workport is 0 bar.

A constant operator signal in the case without disturbance corresponds to a constant spool opening and thus a constant flow into the attached volume. For the backhoe this corresponds to trying to move the boom when it is not possible to move, for example because the crane has hit the ground and cannot be moved further. In the case with disturbance a constant operator signal instead corresponds to moving the boom on the backhoe with a constant velocity. Using a constant operator signal in the simulations makes it easy to see how the different controllers behave and it can be said to have the same functionality as a step response when analysing a regular control algorithm. In this particular case it is easy to identify for example the pressure when the control algorithms starts to decrease the flow from the valve and thus the pressure when the user is overridden by the control algorithms and how the valve closing behaves.

Besides using graphs for comparing the different control algorithms a few performance metrics have been calculated in order to get numerical values on the performance. The first metric is $p_{override}$ which is the pressure when the operator is overridden by the control algorithm. The time from $p_{override}$ until the pressure has reached within 5 bar from the maximum allowed pressure is called T_{final} and this together with $p_{override}$ is a measure on how much the operator is limited by the software pressure feed reducer. If for example $p_{override}$ is very low compared to the maximum pressure and T_{final} is very long, the operator is limited a lot and the

movement of the crane on the backhoe may feel slow and unresponsive. To get a measure on how precise the controllers are and how close the limit the pressure is allowed to be, p_{end} and e_{end} are used. p_{end} and e_{end} are the pressure and the relative error in the end of the simulation, respectively.

Each controller has the same tuning parameters for the cases with and without disturbance. This means that there is a compromise in the behaviour and performance between the two simulation cases since all controllers are not possible to tune to work satisfactory in both cases. This can be seen in Figure 5.1 and 5.2 where the workport pressure, control signals and the spool position for each controller are plotted for the case without disturbance and with disturbance respectively. The performance metrics corresponding to these two simulation cases can be seen in Table 5.1 and 5.2 respectively.

Table 5.1: Performance metrics from simulations without flow disturbance.

	$p_{override}$ [bar]	T_{final} [ms]	p_{end} [bar]	e_{end} [%]
LQR	65	103	99	0.9
LQRI	65	76	104	3.6
\mathcal{H}_∞	93	3	109	9.0
\mathcal{H}_∞ Loop-Shaping	65	164	98	1.7
MPC	58	123	99	0.6
Bumpless PID	65	66	106	6.4

Table 5.2: Performance metrics from simulations with flow disturbance.

	$p_{override}$ [bar]	T_{final} [ms]	p_{end} [bar]	e_{end} [%]
LQR	70	∞	86	14.3
LQRI	70	139	100	0.0
\mathcal{H}_∞	86	197	92	8.1
\mathcal{H}_∞ Loop-Shaping	64	∞	80	20.0
MPC	64	∞	92	8.0
Bumpless PID	64	137	100	0.0

From the simulations it can clearly be noted that the overall behaviour of all controllers is relatively similar. This is especially noticeable for the case without any disturbance seen in Figure 5.1, with the exception of the \mathcal{H}_∞ controller that almost closes the valve fully in one sample. This can also be seen in Table 5.1 and 5.2 where the override pressure $p_{override}$ is similar for all controllers, again with the exception of the \mathcal{H}_∞ controller which overrides the user at a significantly higher pressure compared to the other controllers. Possible reasons for the \mathcal{H}_∞ controller to behave different compared to the other controllers will be discussed later in Chapter 8. Due to the bad behaviour of the \mathcal{H}_∞ controller it will not be implemented and tested on the backhoe, and in the remainder of this section it is not further discussed.

5. Simulation of Controllers

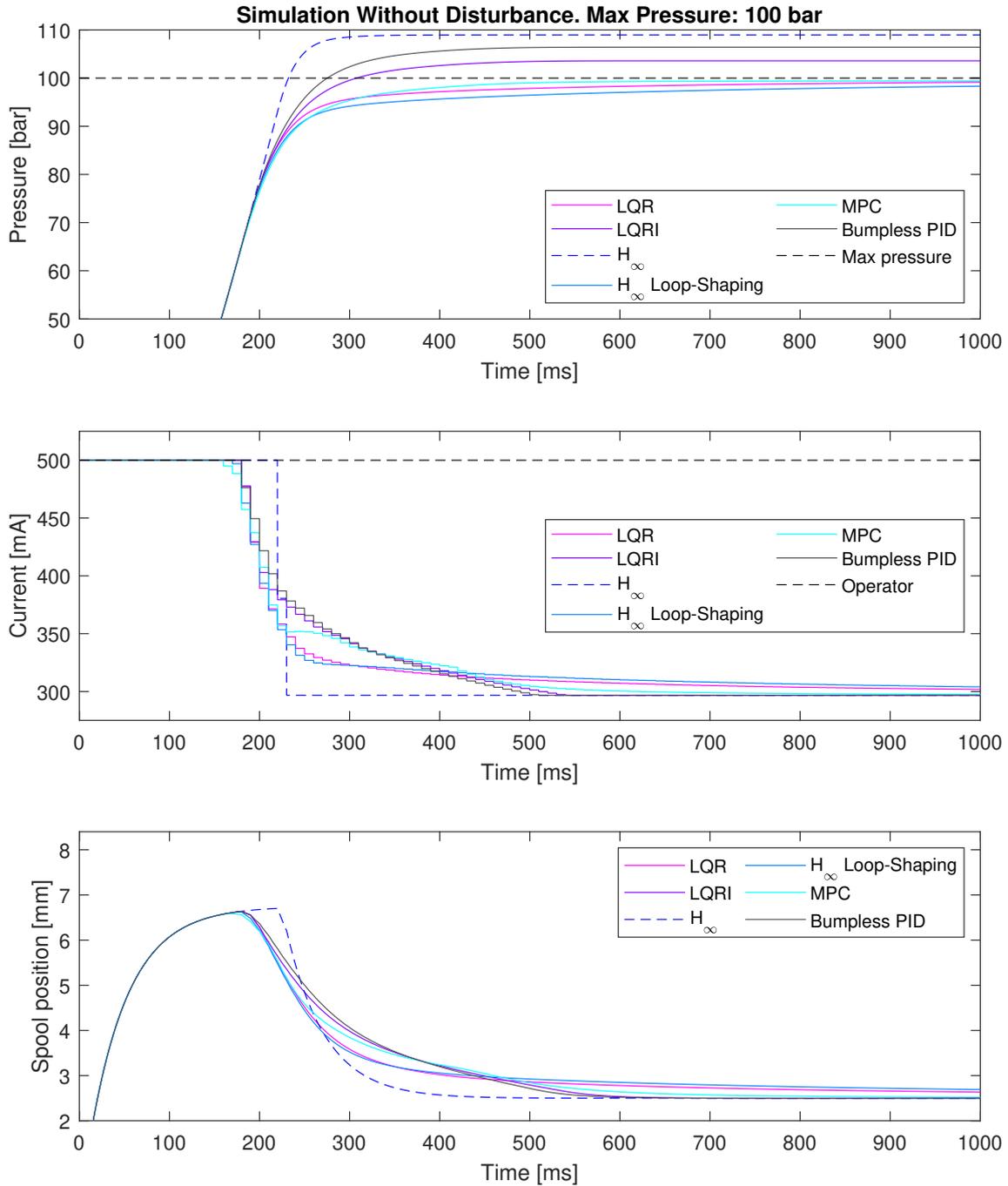


Figure 5.1: Results from simulations without flow disturbance.

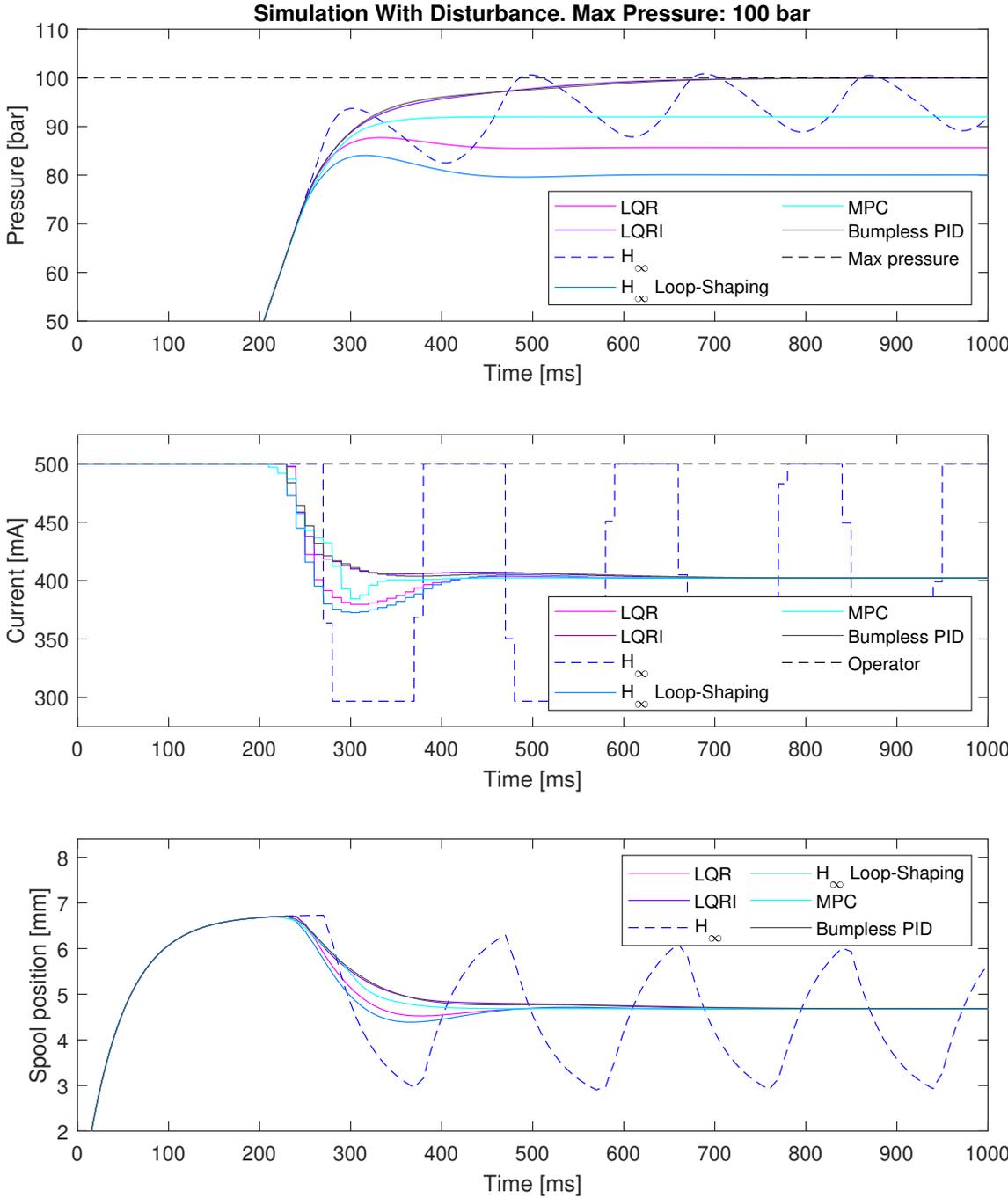


Figure 5.2: Results from simulations with a flow disturbance as a negative pressure derivative of -157 bar/s.

When comparing the simulations with and without the added disturbance it can be noted that the controllers with integral action, i.e. the LQRI and the bumpless PID, overshoot significantly for the case without the added disturbance while the controllers without integrals reach the correct pressure in this case. On the other hand, when the disturbance is added the controllers with integrals reach the correct pressure, while the ones without does not. The reason for the controllers with the integrals to overshoot when the disturbance is not added is that the model is correct in this case, and therefore the controllers would reach the maximum pressure without the integrals. When adding the integrals, they will keep counting up the control signal until the maximum pressure is reached, and therefore make the controllers close the valve too slow, making it overshoot. In the case with the disturbance the model is not correct since it says that the pressure will keep rising as long as the valve is opened, which does not happen when the disturbance is added. Therefore, the integrals are needed to compensate for this disturbance. The conclusion of this is that different tunings are necessary for the controllers to perform well in both the case with and without disturbance.

The controllers that successfully reach the maximum pressure are the LQR, MPC and \mathcal{H}_∞ Loop-Shaping for the case without disturbance and the LQR with integral action and PID with bumpless transfer for the case with disturbance. From Figure 5.1 and 5.2 it can be seen that they reach a pressure level of around 90 to 95 bars quickly, and then slowly approaches the maximum pressure from there. This is due to the nonlinear area curve in the simulation model that is linearized in the model used in the controllers. As can be seen in the area curve in Figure 3.6 the nonlinear curve is flat where the valve starts to open, while the linear approximation is not. This makes the controller use less control signal than it should when the valve is almost closed in order to keep the pressure increasing as fast as expected.

One final note to make from the simulations is that the spool position seen in the figures is quite smooth for all well-behaving controllers. This is important for the implementation on the backhoe since spikes or sharp edges in the spool position would reflect badly on the behaviour of the machine which would be uncomfortable for the operator.

6

Implementation

In order to evaluate the control algorithms for the software pressure feed reducer on the backhoe they have to be implemented on the controller module mentioned in Section 1.7. In this chapter the implementation is briefly described together with signal filtering that is necessary for the software to work properly.

6.1 Controller Implementation

During implementation of the controllers, the same Simulink models of each controller as in the simulation in Chapter 5 were used to generate C-code using the IQAN Simulink Toolbox. The code generated from the toolbox was imported into the IQANdesign software used to program the controller module and the touch display described in Section 1.7 [26].

In the IQANdesign software the Simulink models can be used as function blocks together with regular programs developed directly in IQANdesign [26]. In this way the control algorithms were integrated into the already existing program that controls the majority of the functions on the backhoe. Also, signal processing such as filtering of the measured pressures and comparing the controller signal to the operator signal were implemented directly in IQANdesign. Finally, to be able to faster test different tuning parameters most of them were made adjustable on the touch display in the backhoe.

6.2 Measurement Filtering

The developed control algorithms rely on the pressure measurements from the work-ports. However, the raw measurement signal cannot be used directly since it contains far too much noise and it therefore needs to be filtered before it can be used. One way to filter the pressure signals in the software is to use a low pass filter of low order, but there is a drawback with a simple low pass filter. When filtering the signal enough to get rid of the noise, a delay is introduced for fast changes in the signal and since delays are not desirable for control systems there are better filters to use than a low pass filter of low order.

As mentioned in Section 6.1 the software IQANdesign is used for implementation and in this software there is a filter called band width limiting (BWL) filter that can be used [26]. This filter is adaptive and how much the signal is filtered depends on how much the signal changes between two samples and on the BWL filter parameter. The lower the signal change is, the more the signal is filtered and the bandwidth of the signal is limited more. If the change on the other hand is larger than the BWL filter parameter, the signal is not filtered. This means that if the filter is correctly tuned for the signal to be filtered, fast changes can be captured without delays and noise can still be removed when slower changes occur in the signal.

The same behaviour as the BWL filter is most probably possible to obtain with a low pass filter of higher order where the crossover frequency is variable depending on the signal derivative. However, since the BWL filter was available in the IQANdesign software and it turned out to perform well it was decided to use it instead of implementing a new filter and thereby focus more on developing the control algorithms.

6.3 Input Current Filtering

While testing the software pressure feed reducer a problem with oscillations occurred in some cases, especially when performing a boom lift with high speed. These oscillations always had approximately the same frequency, did not dampen out and occurred due to the control signal oscillating. One way to damp out the oscillations was found to be to filter the current from the controller through a notch filter with a notch frequency set to the frequency of the oscillations as seen in Figure 6.1. This successfully damped out most of the oscillations, and also made the control signal smoother.

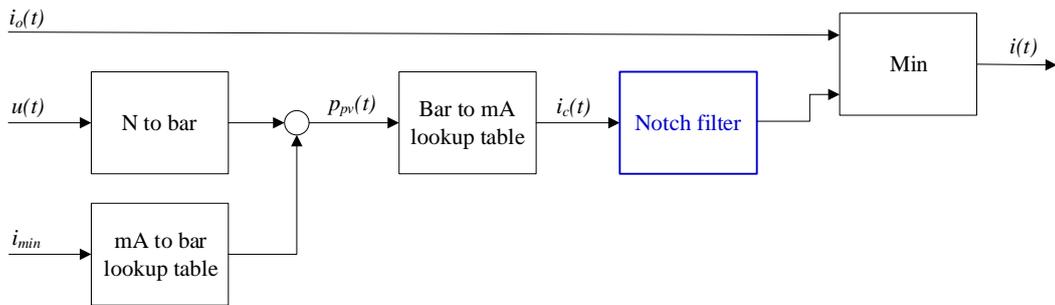


Figure 6.1: Block diagram showing the controller output conversion from Figure 4.10 together with the notch filter depicted in blue.

A notch filter is a type of band stop filter that removes a certain frequency from a signal. The transfer function of a standard notch filter can be written on the form

$$N(s) = \frac{s^2 + \omega_0^2}{s^2 + \omega_c s + \omega_0^2} \quad (6.1)$$

where ω_0 is the notch frequency that the filter removes and ω_c is the width of the notch [27].

Adding the notch filter to the control signal can reduce the oscillations since it stops the controllers from exciting the system in the oscillation frequency, however it has some drawbacks. Since it stops the controller from using the oscillation frequency, it also inhibits it from being able to actively damp out the oscillations. Also, since the notch filter is not ideal it filters out other frequencies near the oscillation frequency. This can reduce the performance of the controllers, and it also removes the theoretical properties of the controllers, such as the guaranteed optimality of the LQR controller. Therefore, it is desired to make the notch as narrow as possible, while still capturing the oscillation frequency.

In order to apply the notch filter in the control module it had to be discretised. This was done using the bilinear transformation commonly used for filters [28]. First the frequencies of interest, i.e. the notch frequency in this case, are pre-warped according to

$$\omega_d = \frac{2}{T_s} \arctan\left(\frac{\omega_a T_s}{2}\right) \quad (6.2)$$

where ω_d is the pre-warped frequency in discrete time corresponding to ω_a in continuous time and T_s is the sampling time [28]. Then the bilinear transformation is applied by changing the Laplace-variable s in the continuous transfer function according to

$$s = \frac{2(1 - z^{-1})}{T_s(1 + z^{-1})}. \quad (6.3)$$

Adding only a notch filter to the controller removed the oscillations, but it also removed the first closing ramp in the signal to the pilot valve. This resulted in the pressure exceeding the maximum pressure since the valve did not close as fast as the controller wanted. To fix this problem, logic to when to turn on and off the notch filter was added. For the filter to be active the measured pressure must be close enough to the maximum pressure and the pressure derivative has to peak and start to decrease. This means that the filter will not be activated until after the controller has started to close the valve in order to decrease the pressure and the first closing ramp in the signal is not removed by the filter. Thus, the pressure will not exceed the maximum pressure and the filter will still be activated early enough to remove all oscillations.

To deactivate the filter a timer is used to keep track of how long time the operator has controlled the valve, i.e. for how long time the controller output has been lower than the operator signal. If the operator has controlled the valve longer than a given threshold value, the notch filter is deactivated and the procedure described above has to be repeated for the filter to be reactivated. The threshold value prevents the filter from being repeatedly activated and deactivated if the oscillations are close to the operator signal and thus the oscillations will be removed by the filter.

6.4 Variable Integral Gain

The PID controller with bumpless transfer and the LQR controller with integral action both have the integral gain K_i as tuning parameter. During testing of these two controllers on the backhoe it turned out that it was not possible to get a satisfactory result with only a constant integral gain as tuning parameter. With a higher K_i gain the maximum pressure was exceeded with a big overshoot and with a K_i gain low enough to not exceed the maximum pressure the operator was limited to much and the system was very slow. Therefore, a variable K_i gain that changes depending on the offset from the maximum pressure was introduced. When the measured pressure is far from the maximum pressure the integral gain is kept high in order to get a fast responding system, but when the pressure is getting closer to the maximum level the gain is decreased in order to prevent it from exceeding the maximum pressure. This turned out to perform well and made it possible for the software pressure feed reducer to limit the operator less and still not exceed the maximum pressure.

The variable K_i gain was implemented as a lookup table between the offset from the maximum pressure and the integral gain K_i . Linear interpolation was used between the table elements.

7

Results

In this chapter results from tests of the different control algorithms implemented on the backhoe are presented. Three different scenarios have been tested and these are described together with the system limitations for the backhoe in order to have a better understanding for the results. Graphs and performance metrics are presented for the three different scenarios and for each controller presented in Chapter 4, except the \mathcal{H}_∞ controller.

The controllers have been tuned to obtain as good behaviour as possible in each case which means that the tuning parameters for each controller varies between the different scenarios. To obtain a good behaviour a compromise had to be done between a smooth behaviour in the backhoe and how much the operator is limited by the control algorithm.

From all tests, data was only available approximately every 60 ms since the CAN-bus used for logging was heavily congested and many samples were therefore missing. Due to this some of the figures presented in this chapter look a bit spiky and not that smooth. However, even though there are samples missing during the logging, the control signals to the pilot valves are sent and received every 10 ms since it is implemented directly on the controller module. Thus, this only affects logged data and not the controller performance.

7.1 Test Description

Three different test cases were performed on the backhoe, namely a boom down case, a boom lift case and a digging case. In the boom down case, the bucket on the backhoe was placed on solid ground, and then it was pushed downwards against the ground using the boom function in the direction depicted by 1 in Figure 7.1. This direction uses workport B on the valve, and a maximum pressure of 90 bar was used. This maximum pressure was chosen since it is not enough to lift the backhoe, and therefore the system behaves similar to the model used in the controllers due to the cylinder not moving significantly. Only one pressure level is presented here since using other pressure levels that are low enough to not lift the backhoe yielded very similar results.

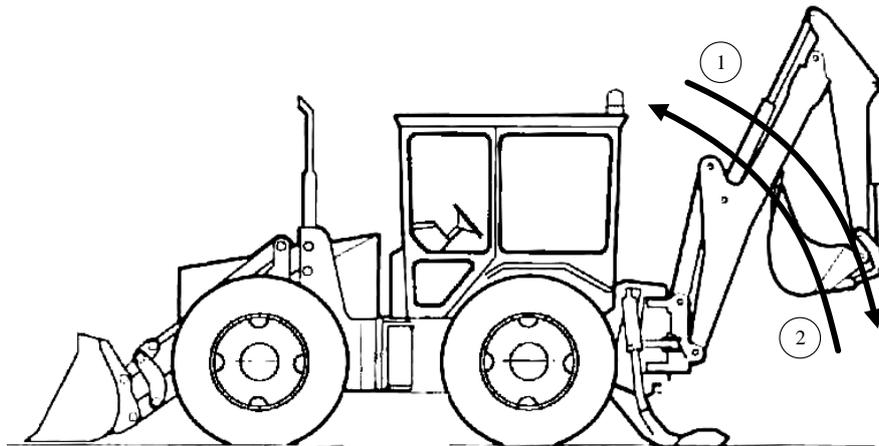


Figure 7.1: Schematic description of the boom lift and boom down test cases. 1. Boom down, 2. Boom lift. From [7]. Adapted with permission.

In the boom lift case the the bucket was placed on the ground, and then the boom was lifted in the direction depicted by 2 in Figure 7.1 which uses workport A on the valve. When performing this test the boom moves upwards, and therefore this test corresponds to the simulations in Section 5.2 with the added disturbance. To properly test this, two different pressure levels were used, 90 bar and 110 bar. At 90 bar the boom only moves slowly upwards while at 110 bar it moves significantly faster. One additional test was performed in the boom lift test case, namely one with a maximum pressure of 110 bar and a lower operator signal. This test was added to show how well the controllers perform when the operator should not be limited by the controller in steady state since at this operator signal the pressure does not exceed 110 bar in steady state. The operator input was for this test 60 %, where 0 % represents the minimum current and 100 % represents the maximum current, i.e. the current required to open the valve fully.

The final test case is a digging case. This was done in the same direction as the boom lift, i.e. using workport A on the valve. The difference is that in this case the bucket was not resting on the ground when starting the test, rather it was used for digging. This means that the bucket was filled with gravel that was heavy enough for the maximum pressure used to not be able to lift it. This way the boom was not moving significantly during the test, and therefore the behaviour of the system is similar to the model used in the controller designs. Again, both 90 bar and 110 bar maximum pressure were tested, although only at 100 % operator input.

During all tests the operator signal was generated as a ramp of 200 ms from the minimum current to the set operator input to get repeatable results. A ramp was used rather than a step to protect the machine.

7.2 System Limitations

In order to evaluate the performance of the controllers on the physical system, the system behaviour and limitations had to be investigated. Most importantly, a baseline for how fast it is possible to stop the pressure from rising in different scenarios is needed. In order to investigate this, the valve was closed fully at a pressure level a certain amount below the maximum limit. The pressure offset from the maximum limit was chosen such that pressure did not increase above maximum pressure. The resulting values can be seen in the column Close Pressure Offset in Table 7.1. In the same table the time from closing the valve until the pressure stopped rising can be seen in the column Closing Time. Note however that the closing time is approximate due to the sampling time of the logging as mentioned before. The metrics seen in the table are useful when tuning and evaluating the controllers, since they show the fastest way to close the valve and therefore the physical limitations of the system. It is not possible to start closing the valve at a pressure closer to the maximum pressure than the close pressure offset in the table for a certain case. It is also not possible to stop the pressure from exceeding the maximum pressure faster than the closing time in the table.

Table 7.1: Table showing the system limitations obtained by closing the valve as fast as possible. 60 % and 100 % are the operator signals, and 90 bar and 110 bar are the maximum pressures.

	Close Pressure Offset [bar]	Closing Time [ms]
Boom Down, 90 bar, 60 %	5	60
Boom Down, 90 bar, 100 %	20	60
Boom Lift, 90 bar, 60 %	38	120
Boom Lift, 90 bar, 100 %	60	140
Boom Lift, 110 bar, 100 %	60	120
Digging, 90 bar, 60 %	38	140
Digging, 90 bar, 100 %	60	120
Digging, 110 bar, 60 %	38	120
Digging, 110 bar, 100 %	60	120

What can be seen in the table is that, for the boom down cases, the valve does not need to be closed as early as for the boom up and digging cases. The reason for this is that the area curves for the spool used in the implementation is significantly flatter for workport B, which is used for boom down, compared to workport A. This means that the flow is significantly smaller in port B, which in turn makes the pressure increase slower.

Something else that can be noted from the table is that both tests that are performed on workport A, i.e. boom lift and digging, can use the same pressure offset for the same operator input. For example, for 60 % operator signal the valve needs to be closed at 38 bar before the maximum pressure is reached, no matter if doing a boom lift or digging, or if the maximum pressure is set to 90 bar or 110 bar.

The behaviour from closing the valve as fast as possible can be seen in Figure 7.2 when performing a boom lift with a maximum pressure of 110 bar, an operator signal of 100 % and closing the valve 60 bar before reaching the maximum pressure. The behaviour is similar for the other cases in Table 7.1. Two things are worth noting from the figure. First, no oscillations can be seen in the pressure. This is however not true in reality. The crane on the backhoe is oscillating, which causes pressure oscillations. The reason they do not show in the figure is that a hose burst valve is used on the boom cylinder. This is used to stop the boom from falling down in case of a hose burst and blocks the oil between the hose and cylinder when it is not actively driven. This means that when the valve is closed the pressure oscillations caused by the load swinging only occur in the cylinder, and not in the hose and valve. Therefore, the pressure sensor used will not register the oscillations since it is mounted close to the valve, and they will not be visible to the controller or in the figures. This will however not cause any major problems for the control algorithms since it only comes into effect when the valve is already closed. Even though these oscillations do not appear in the figure, they are not desired and cause wear on the machine. Therefore, closing the valve this fast is not desired.

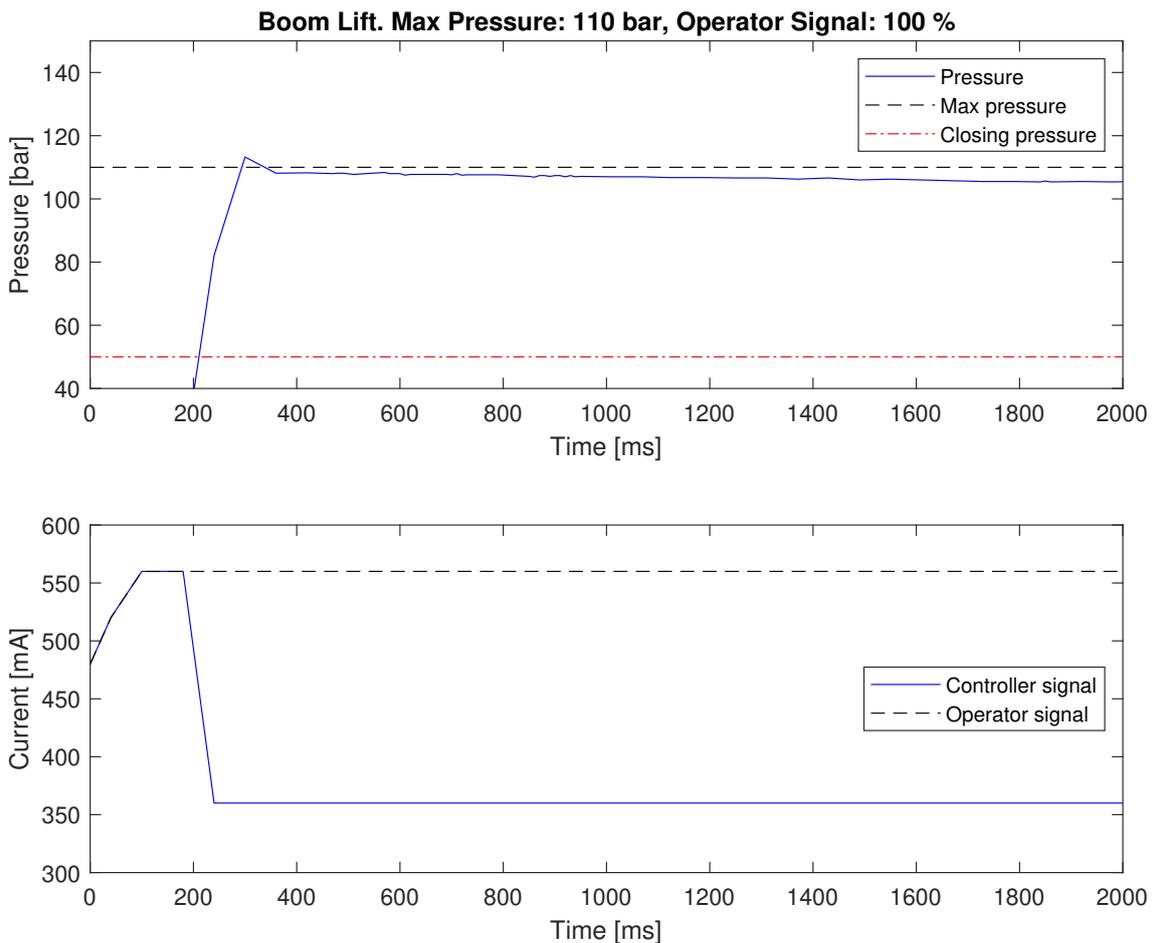


Figure 7.2: Behaviour from closing the valve as fast as possible 60 bar below the maximum pressure of 110 bar during a boom lift with 100 % operator signal.

The second thing to notice from the figure is that the pressure slowly decreases after the valve is closed. This happens due to oil leakage in the valve, and helps in case the pressure overshoots since it makes it possible to reduce the pressure. However, even with this fact, overshoot is not desired since the pressure should always be kept below the maximum limit. Also, closing the valve fully to reduce the pressure might give rise to other problems due to for example the hose burst valve mentioned above.

7.3 Controller Comparison

In order to evaluate the performance of the controllers implemented on the backhoe the same performance metrics that were used for the simulation are used, but with some small changes. p_{end} and e_{are} are not the pressure and the relative error at the end of the simulation respectively, rather they are the pressure and the relative error at the end of the sequences presented in the figures. Beyond these performance metrics, one additional metric called p_{peak} is used. This metric is defined as the maximum pressure during the logged sequence.

For the boom lift test case with 110 bar maximum pressure and 60 % operator signal all the performance metrics mentioned above are not used. Instead only $p_{override}$ is used together with $T_{overridden}$ which is the time in milliseconds the user is overridden by the control algorithm.

Graphs from the boom down test case with 90 bar maximum pressure and 100 % operator signal can be seen in Figure 7.3 and performance metrics corresponding to this test sequence can be seen in Table 7.2. From the figure it can be seen that the control signal for the MPC controller is very spiky which resulted in noticeable vibrations in the backhoe. Also, the control signal for the \mathcal{H}_∞ Loop-Shaping controller is quite spiky, but almost no oscillations occurred in the pressure and no vibrations were noticeable in the machine.

Something to note from the same figure and table is that the LQR with integral action is not present. This is due to the integrator part giving rise to an overshoot in the pressure and the integrator has to be removed in order to not exceed the maximum pressure. Thus, it is the same controller as the LQR and only plotted once. Another thing that can be seen from the figure is the same creeping pressure behaviour that was seen in the simulation in Section 5.2. This is especially noticeable for the LQR controller and the PID controller with bumpless transfer where the pressure increases slowly when it is getting close to the maximum pressure.

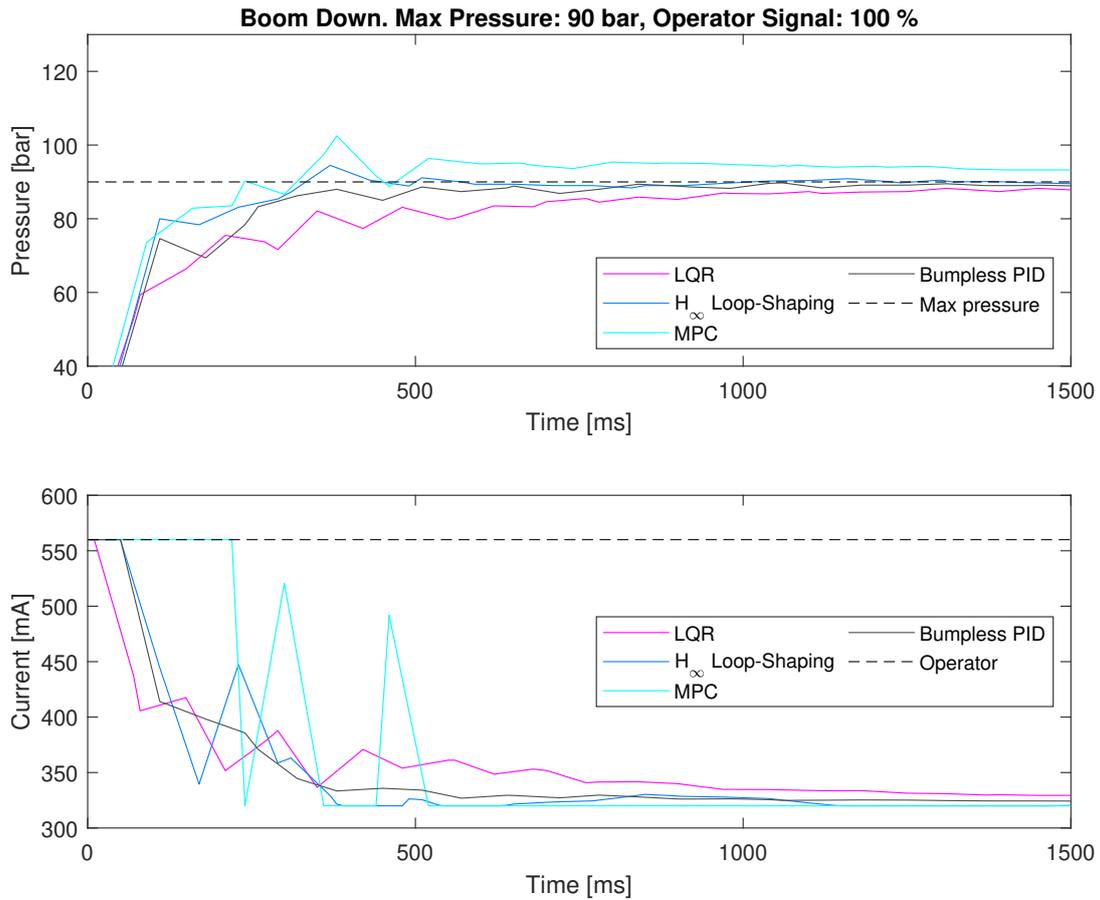


Figure 7.3: Results from boom down test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.

Table 7.2: Performance metrics for boom down test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.

	$p_{override}$ [bar]	T_{final} [ms]	p_{peak} [bar]	p_{end} [bar]	e_{end} [%]
LQR	29	740	88	88	2.6
\mathcal{H}_∞ Loop-Shaping	65	240	93	90	0.1
MPC	87	161	100	94	4.2
Bumpless PID	41	270	89	89	1.1

Graphs from the boom lift test case with 90 bar maximum pressure and 100 % operator signal can be seen in Figure 7.4 and performance metrics corresponding to this test sequence can be seen in Table 7.3. A couple of things can be noted from this test. First, both the MPC and the \mathcal{H}_∞ Loop-Shaping overshoot significantly. Second, the LQR with integral action and PID with bumpless transfer both reach the maximum pressure, but they end up at different control signals in steady state. Finally, the LQR controller does not manage to reach the maximum pressure.

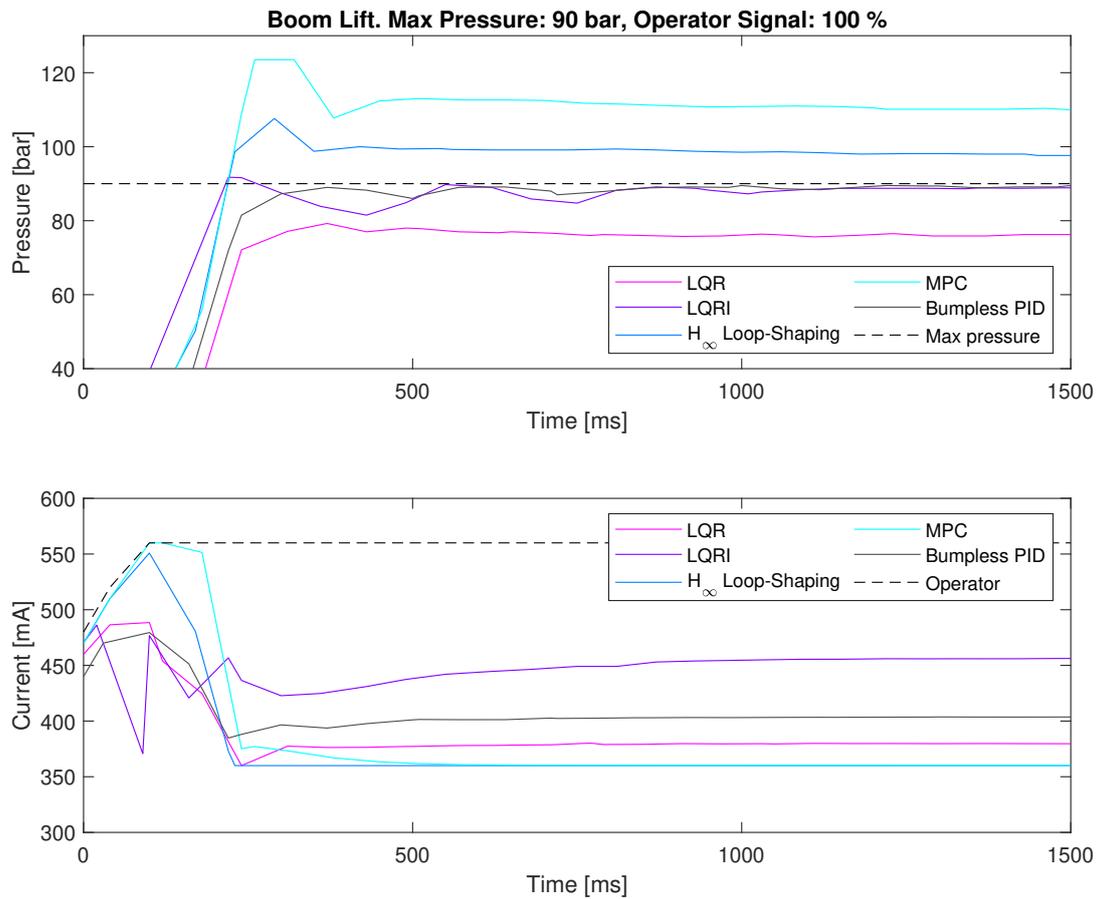


Figure 7.4: Results from boom lift test case with 90 bar maximum pressure and 100 % operator signal. Notch filter enabled.

Table 7.3: Performance metrics for boom lift test case with 90 bar maximum pressure and 100 % operator signal. Notch filter enabled.

	p_{override} [bar]	T_{final} [ms]	p_{peak} [bar]	p_{end} [bar]	e_{end} [%]
LQR	11	∞	78	76	15.4
LQRI	18	210	90	89	1.5
\mathcal{H}_∞ Loop-Shaping	22	110	109	98	9.0
MPC	54	220	125	110	22.4
Bumpless PID	12	270	89	89	1.0

7. Results

Graphs from the boom lift test case with 110 bar maximum pressure and 100 % operator signal can be seen in Figure 7.5 and performance metrics corresponding to this test sequence can be seen in Table 7.4. Note that in Table 7.4 T_{final} for the PID controller is significantly higher than for the other controllers. This is due to the performance metric not capturing the behaviour well in this specific case. Again, the MPC and the \mathcal{H}_∞ Loop-Shaping controller overshoot the maximum pressure. It can also be noted from the figure that some oscillations occur in the pressure for the LQR with integral action.

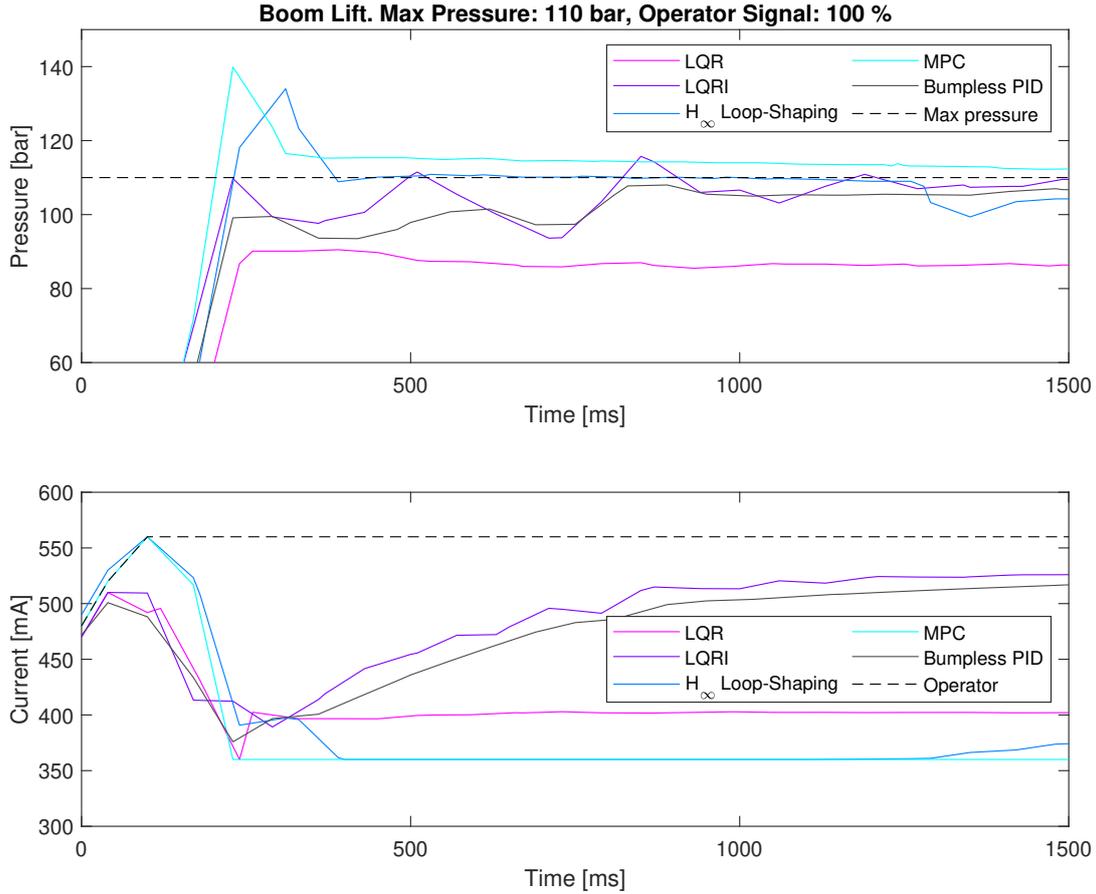


Figure 7.5: Results from boom lift test case with 110 bar maximum pressure and 100 % operator signal. Notch filter enabled.

Table 7.4: Performance metrics for Boom lift test case with 110 bar maximum pressure and 100 % operator signal. Notch filter enabled.

	$p_{override}$ [bar]	T_{final} [ms]	p_{peak} [bar]	p_{end} [bar]	e_{end} [%]
LQR	11	∞	90	86	21.4
LQRI	9	180	114	108	1.5
\mathcal{H}_∞ Loop-Shaping	38	110	136	103	6.1
MPC	60	210	138	113	2.6
Bumpless PID	10	830	108	106	3.3

Graphs from the boom lift test case with 110 bar maximum pressure and 60 % operator signal can be seen in Figure 7.6 and performance metrics corresponding to this test sequence can be seen in Table 7.5. From the figure and table it can be noted that the LQR, the MPC and the \mathcal{H}_∞ Loop-Shaping controller limit the operator too much, although the MPC is not as limiting as the LQR and the \mathcal{H}_∞ Loop-Shaping controller.

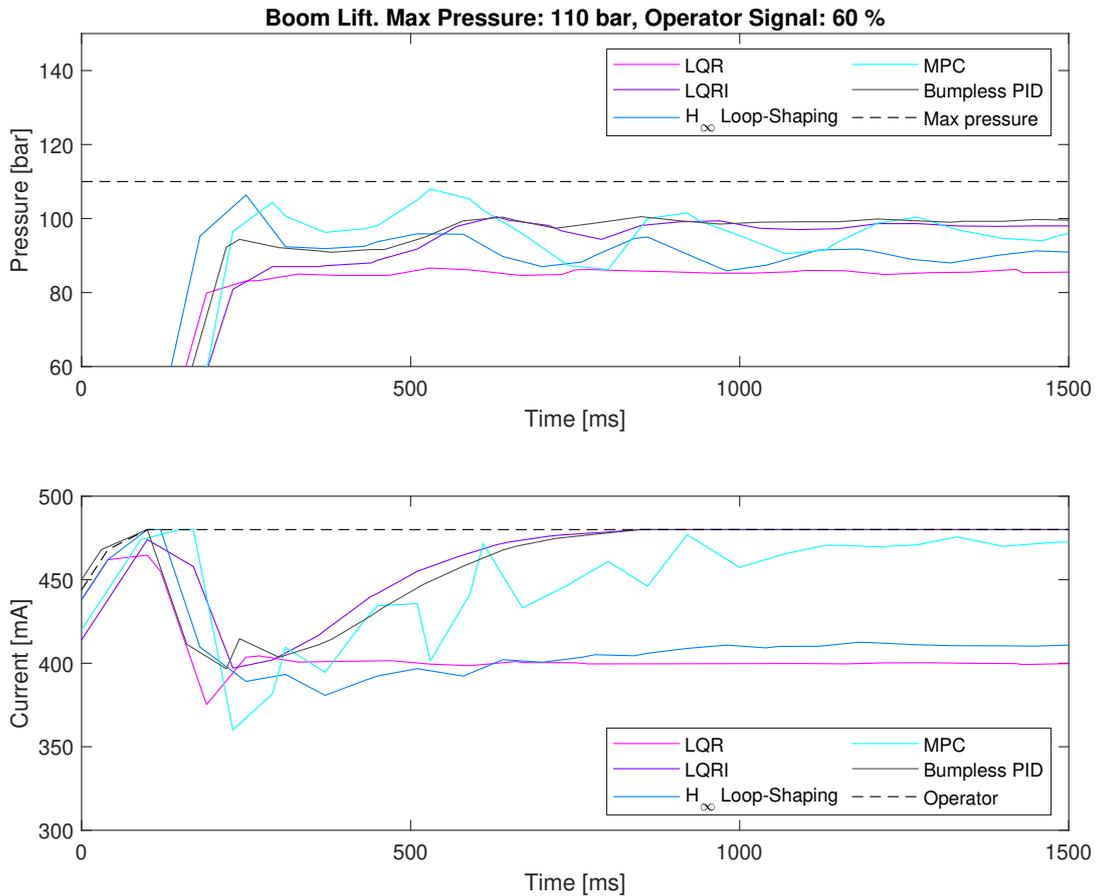


Figure 7.6: Results from boom lift test case with 110 bar maximum pressure and 60 % operator signal. Notch filter enabled.

Table 7.5: Performance metrics for boom lift test case with 110 bar maximum pressure and 60 % operator signal. Notch filter enabled.

	$p_{override}$ [bar]	$T_{overridden}$ [ms]
LQR	23	∞
LQRI	29	630
\mathcal{H}_∞ Loop-Shaping	76	∞
MPC	66	∞
Bumpless PID	26	680

7. Results

Graphs from the digging test case with 90 bar maximum pressure and 100 % operator signal can be seen in Figure 7.7 and performance metrics corresponding to this test sequence can be seen in Table 7.6. Again, the MPC and \mathcal{H}_∞ controllers exceed the maximum pressure. The LQR almost reaches the maximum pressure, and the other controllers reach close to the limit with a good behaviour.

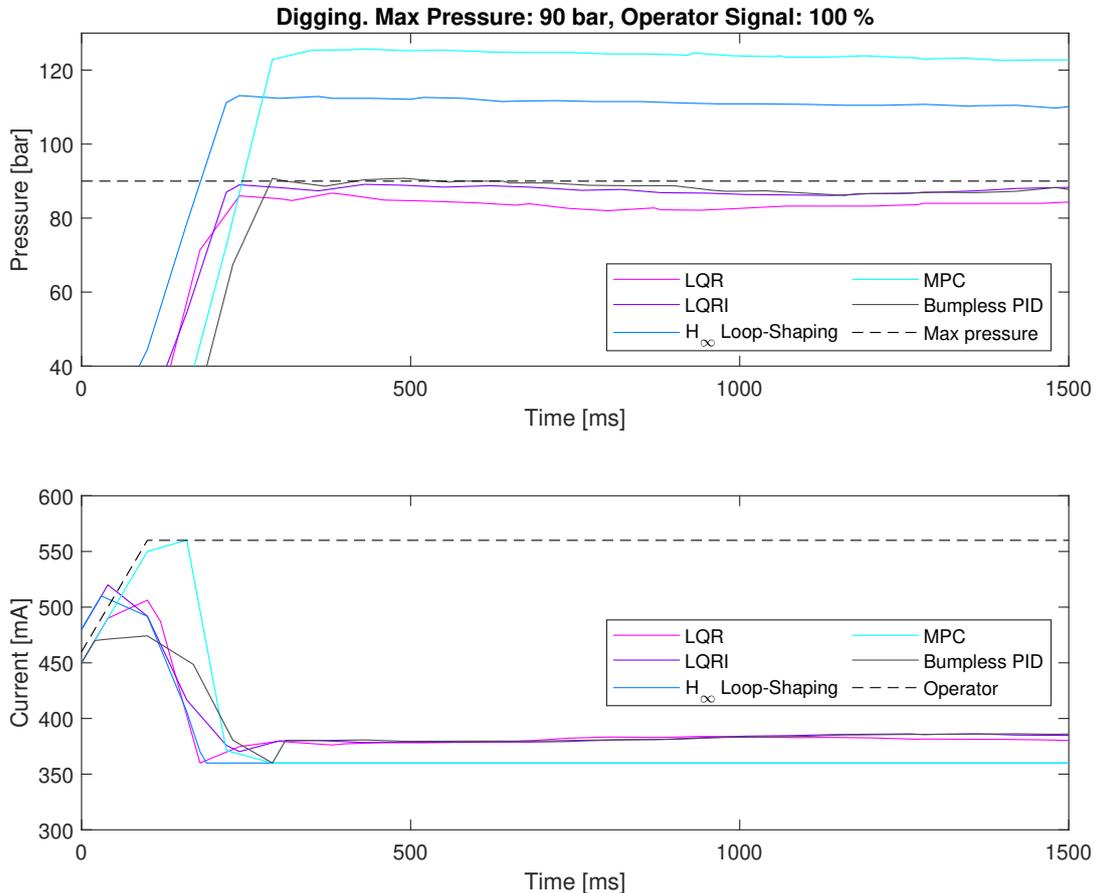


Figure 7.7: Results from digging test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.

Table 7.6: Performance metrics for digging test case with 90 bar maximum pressure and 100 % operator signal. Notch filter disabled.

	$p_{override}$ [bar]	T_{final} [ms]	p_{peak} [bar]	p_{end} [bar]	e_{end} [%]
LQR	12	360	86	84	6.9
LQRI	11	180	89	88	2.6
\mathcal{H}_∞ Loop-Shaping	21	120	112	110	22.6
MPC	40	240	125	123	36.2
Bumpless PID	9	280	90	87	3.1

Graphs from the digging test case with 110 bar maximum pressure and 100 % operator signal can be seen in Figure 7.8 and performance metrics corresponding to this test sequence can be seen in Table 7.7. As for the 90 bar digging case it can be noted that the MPC and the \mathcal{H}_∞ Loop-Shaping controller exceeds the maximum pressure and the rest of the controllers have a satisfactory behaviour and reaches the maximum pressure.

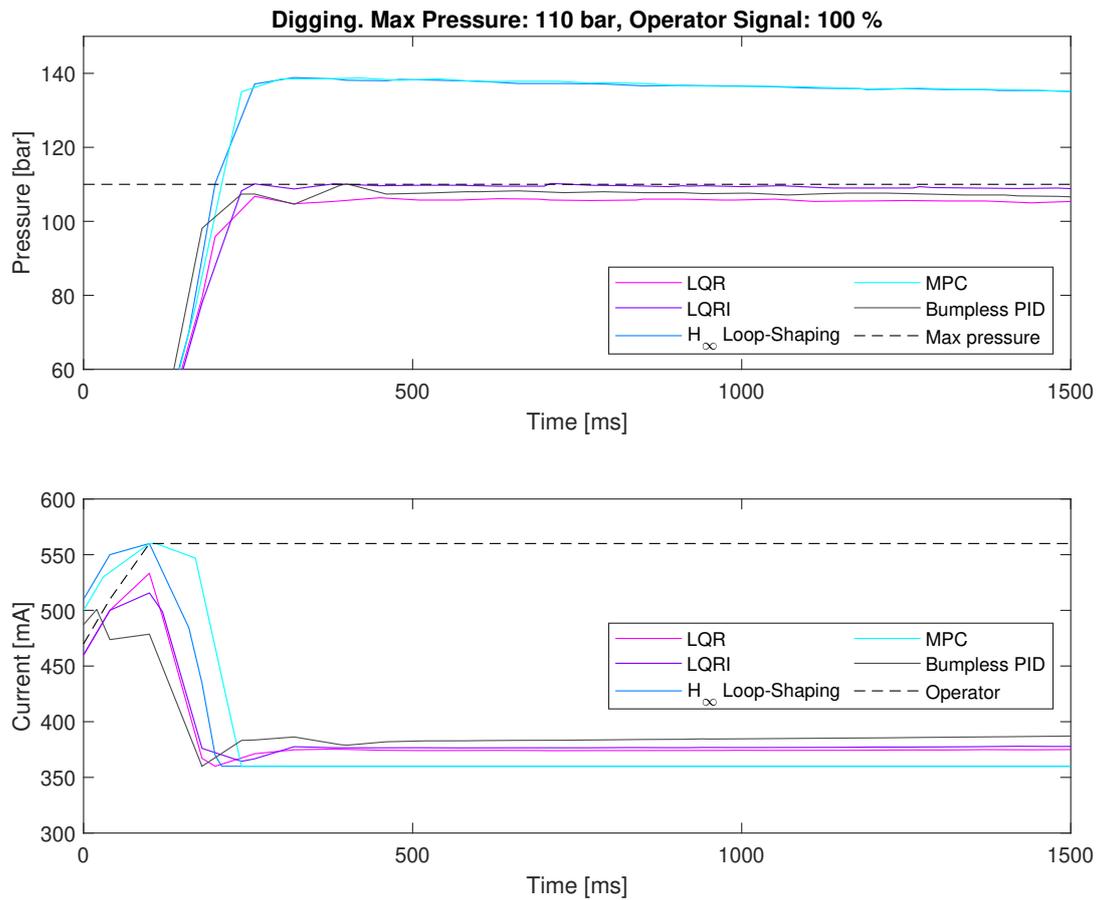


Figure 7.8: Results from digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter disabled.

Table 7.7: Performance metrics for digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter disabled.

	$p_{override}$ [bar]	T_{final} [ms]	p_{peak} [bar]	p_{end} [bar]	e_{end} [%]
LQR	18	210	106	105	4.3
LQRI	18	190	110	109	0.8
\mathcal{H}_∞ Loop-Shaping	38	90	138	136	23.3
MPC	66	210	138	136	23.3
Bumpless PID	9	240	108	107	2.6

7.4 System Phenomena

There are a number of phenomena that were noted during testing of the control algorithms for the software pressure feed reducer. Some of them are properties of the system that the current controllers do not handle well while others, such as pressure oscillations, have been solved. In this section the different phenomena will be presented.

7.4.1 Pressure Oscillations

The pressure oscillations that occurred during implementation without the notch filter mentioned in Section 6.3 can be seen in Figure 7.9 where a boom lift test case with 110 bar maximum pressure and 100 % operator signal have been performed with the bumpless PID controller. This is a typical example on the oscillations that occurred for the controllers without the notch filter and will be further discussed in Chapter 8.

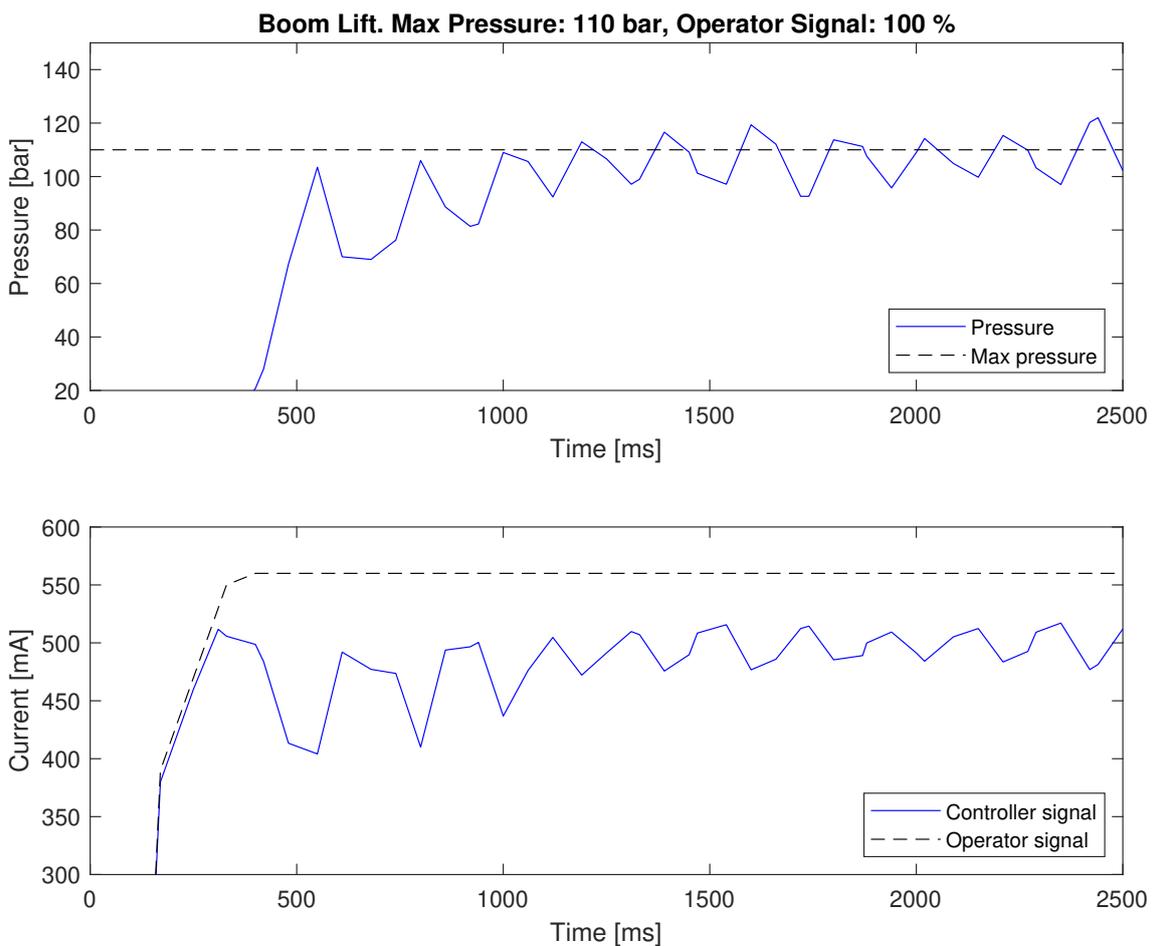


Figure 7.9: Oscillations from PID with bumpless transfer without the notch-filter. Boom lift test case with 110 bar maximum pressure and 100 % operator signal.

7.4.2 Dropping Pressure

Another phenomenon that was seen during testing is a sudden drop in pressure when the valve opens after being closed. This is noticeable when a controller overshoots the maximum pressure. When the pressure then decreases below the maximum pressure due to the leakage in the valve the controller opens it again to compensate. This causes the pressure to drop as can be seen in Figure 7.10 around the time 1000 ms. This figure was generated using the PID controller with bumpless transfer, but the phenomenon occurs for all controllers after overshooting since it is a property of the system and not the controller. Another thing to notice is that around the time 3000 ms the pressure suddenly increases even though the controller has slowly been opening the valve for two seconds.

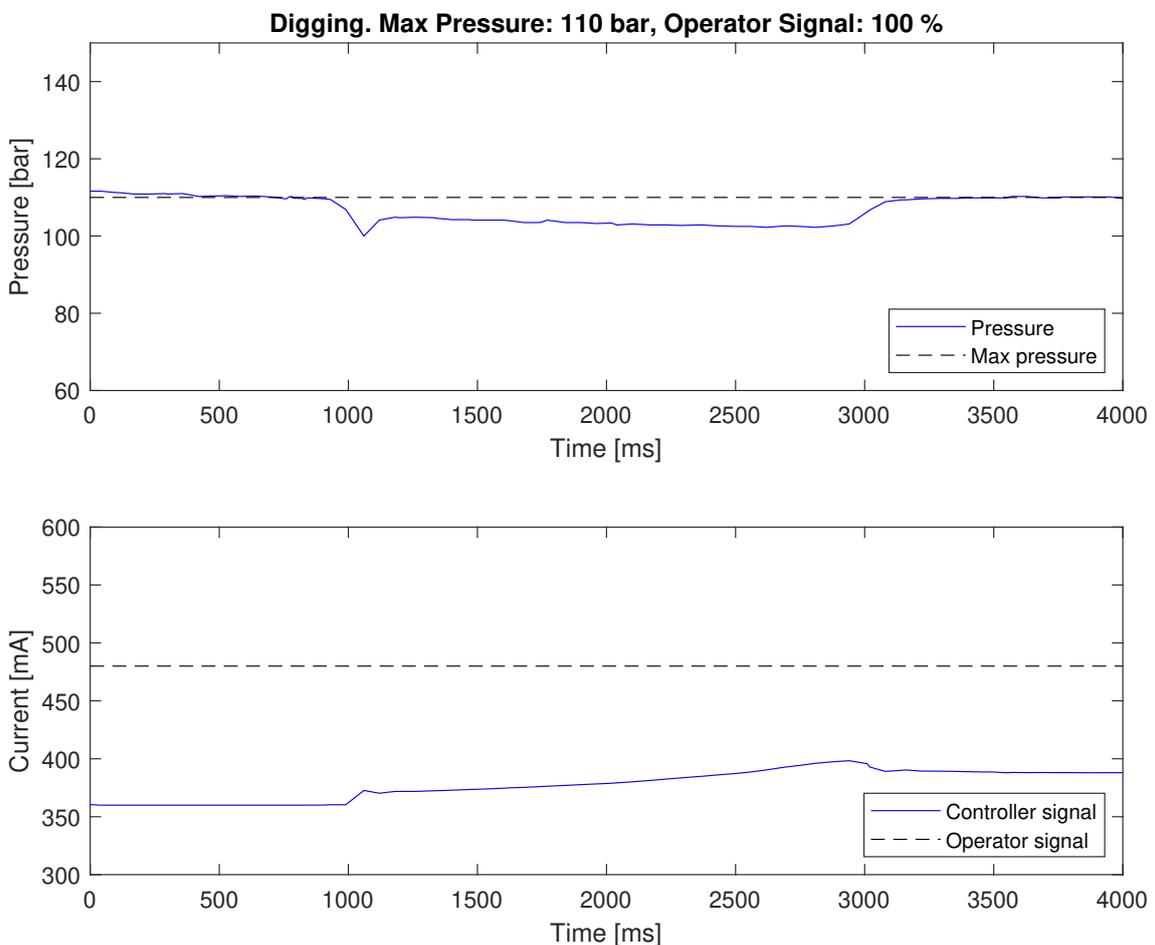


Figure 7.10: Dropping pressure from PID with bumpless transfer. Digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter disabled.

7.4.3 Varying Dynamics

As mentioned in the introduction to this chapter, different tunings were necessary for the three different test scenarios. In order to see how sensitive the controllers are for different tunings and different scenarios the boom lift case was performed with the tunings for the digging case and the digging was performed with the boom lift tunings. Since the LQRI and the PID with bumpless transfer were the only controllers that in all test scenarios reached within 5 bar from the maximum pressure or reached the operator signal, these are the controllers that have been tested with interchanged tunings. However, the result were approximately the same for both controllers and therefore only the result for the PID controller is shown in Figure 7.11 and 7.12. Figure 7.11 shows the boom lift scenario with dig tuning and Figure 7.12 shows the digging scenario with the boom lift tuning.

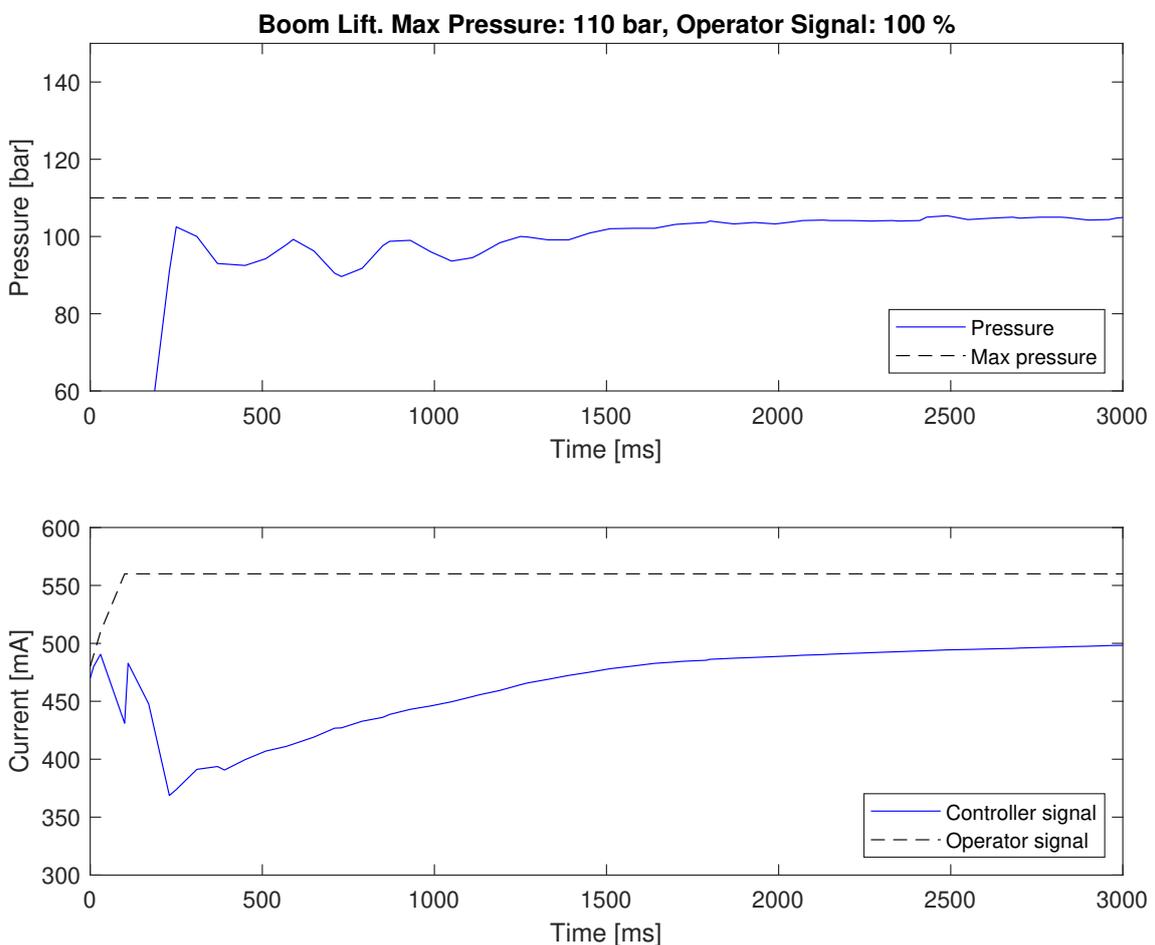


Figure 7.11: Results from PID with bumpless transfer using tunings from digging test case with added notch filter. Boom lift test case with 110 bar maximum pressure and 100 % operator signal.

From Figure 7.11 it can be noted that the operator is limited more than necessary when using digging tuning for the boom lift scenario and from Figure 7.12 it can be seen that the maximum pressure is exceeded when using boom lift tuning for the digging test case.

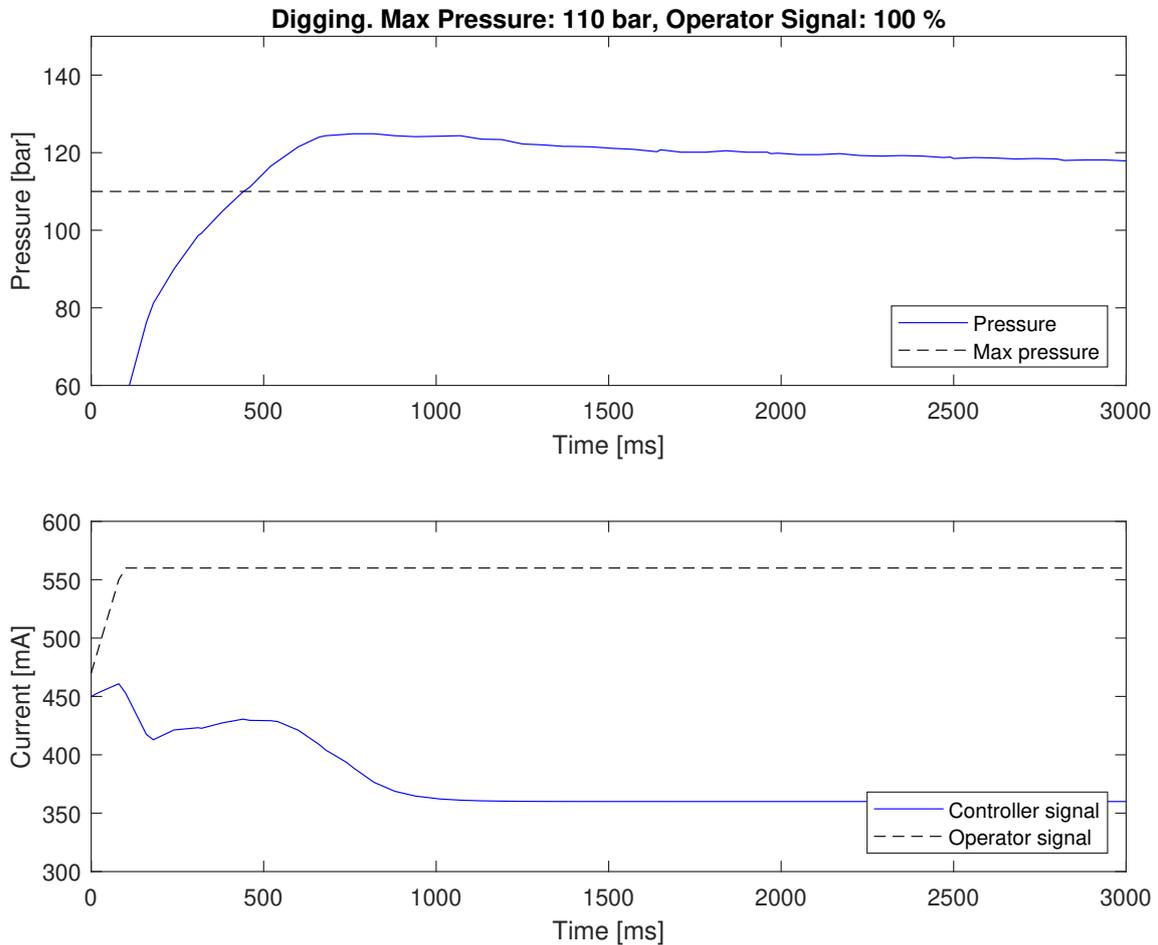


Figure 7.12: Results from PID with bumpless transfer using tunings from boom lift test case. Digging test case with 110 bar maximum pressure and 100 % operator signal. Notch filter enabled.

8

Discussion

In this chapter the results in the previous chapter are discussed. The different controllers are compared, the used valve model is evaluated and the specific system phenomena are further analysed.

8.1 Controller Comparison

The \mathcal{H}_∞ controller designed in Section 4.4 was only tested in simulation and it was not implemented on the backhoe since it did not yield a satisfactory result in the simulations. The \mathcal{H}_∞ controller is not easy and intuitive to tune since there are no specific tuning parameters, rather there are seven different frequency dependent weights to design. The weights should be based on the physical system and for the simulation reasonable parameters for the system were chosen. With these weights the result was basically a controller that just closed the valve in one sample, as can be seen in Figure 5.1. Only changing the weights slightly did not result in a better behaviour and to get a satisfactory behaviour all seven weights most probably need to be changed significantly. However, this is a massive job that takes a lot of time and besides this the main point of using the \mathcal{H}_∞ controller is lost if weights with physical meaning are not chosen [18].

The controller that is obtained when using the \mathcal{H}_∞ controller is not just static gains, it is a state space system with several different states without any physical interpretation. This makes the controller quite unintuitive and it is hard to further develop the control algorithm and base extra functions on this state space system. All this makes the \mathcal{H}_∞ controller unsuitable for the software pressure feed reducer and instead a \mathcal{H}_∞ Loop-Shaping controller was designed and implemented. Since this controller only has two adjustable weights it is easier to tune and not as complex as the \mathcal{H}_∞ controller. This controller gave a satisfactory result in simulation without disturbance as can be seen in Figure 5.1 and was intuitive to tune. However, since there is no integrator in the controller it did not handle the pressure derivative disturbance. Once it was implemented on the backhoe it was only working satisfactory in the boom down test scenario. This might be explained by the model used for the controller not being accurate enough for the boom lift and digging case together with the flow out of workport A being four to five times higher than the flow out of

workport B, as mentioned in Section 2.4. Even if the controller takes into account the different flows for each port it most probably gets more sensitive for model error when the flow is higher. The pressure then increases faster with a lower control signal and the error will be more noticeable in the measured pressure.

Another possible reason for the \mathcal{H}_∞ Loop-Shaping controller not working satisfactorily for workport A and the boom lift and digging test case might be the discretisation of the controller. It is designed in continuous time and then the obtained state space system is discretised using zero order hold. This means that the \mathcal{H}_∞ Loop-Shaping synthesis does not take the sampling time of 10 ms into account and since the dynamics are faster for workport A than B, the controller gets more sensitive for the sampling time on port A. This might explain the worse behaviour during boom lift and digging.

The MPC controller performed well in simulation, but implemented on the backhoe the obtained result was not satisfactory for the three test scenarios, even though it did perform better for boom down than for digging and boom lift. Theoretically this controller should be optimal since it is designed to limit the pressure rather than controlling it, but due to the model not mimicking the physical system perfectly the implemented controller is not optimal. The MPC controller relies heavily on the mathematical model of the system and since the load model is an approximation it affects the final result significantly. Besides this, the controller module used for implementation has limited performance in relation to the high computational complexity of the MPC controller and thus the prediction horizon used is only 70 ms and the maximum number of solver iterations K_{max} is 10. This might reduce the performance of the controller and a longer prediction horizon and more solver iterations might make the MPC yield better results.

As can be seen in Section 5.2 and 7.3 the LQR has a quite satisfactory behaviour, but does not reach the maximum pressure in all test scenarios due to the lack of integrator. This is something that is not possible to affect with tuning since the missing integrator always will make the controller not reach the maximum pressure in situations where an integrator is required. For the boom down case, the model used in the controller reflects the physical system quite good since the cylinder cannot be moved without the pressure increasing. This explains why the maximum pressure is reached for this test scenario, but not in the other ones. When performing boom down it was hard to tune the controller without oscillations and at the same time not limit the operator to much, which explains the oscillations seen in Figure 7.3. The oscillations could be removed with a notch filter as in the boom lift case, but introducing a notch filter for the boom down case made the pressure exceed the maximum pressure.

The LQR with integral action is one of the two best performing controllers since it always reached within 4 bar from the maximum pressure or reached the operator signal, which can be seen in the result graphs in Section 7.3. Even if this controller performs well in most test scenarios, there is one case where the behaviour is not that good. In the boom lift case with 110 bar maximum pressure and 100 % operator signal, seen in Figure 7.5, oscillations on approximately 10-20 bar can be seen in the pressure graph, but almost no oscillations can be seen in the corresponding control

signal. This means that the oscillations are not created by the controller, rather they arise from the machine itself due to closing the valve. Since it was not the controller that created the oscillations they were not as noticeable in the machine as when not using the notch filter. Since the control signal is filtered through the notch filter there is nothing the controller can do about these oscillations and in order to remove them one possibility might be to complement the control algorithm with a solution to remove the last oscillations. One possible algorithm might be the active damping described in [29].

Something that is beneficial for the LQR with integral action is that it is intuitive to tune and it is quite easy to understand and therefore possible to further develop with more functionality.

The last controller that was implemented is the PID with bumpless transfer that performs as good as the LQR with integral action and thus also is one of the two best performing controllers. There is one small difference in the behaviour between the PID and the LQRI controllers. From the tables with performance metrics in Section 7.3 it can be noted that the PID controller in all test cases has a T_{final} that is longer than the one for the LQR with integral action. This means that the PID controller limits the operator more than the LQRI controller, but it also results in a smoother behaviour in the machine, which makes it hard to decide which one is the best.

In order to decide which controller that is best when considering the behaviour and how much the operator is limited, one has to know what a suitable T_{final} is and how the the behaviour in the machine should be. These two properties are closely related to each other and it is hard to specify a desired value on T_{final} since different operators might have different preferences. Therefore, the most suitable alternative would be to have the behaviour of the software pressure feed reducer adjustable such that the operator can chose how fast or limiting the function should be. The PID with bumpless transfer is the controller most suitable for this, since the tuning parameter T_{pred} does exactly this.

Something that the PID with bumpless transfer and the LQR with integral action have in common is that $p_{override}$ is quite low compared to the mechanical pressure feed reducer and thus both these controllers limit the operator quite a lot. As mentioned in Section 2.4 the mechanical pressure feed reducer starts to limit the operator approximately 30 bar below the maximum pressure depending on the flow. The PID and the LQRI controllers start to limit the operator approximately 80 to 100 bar below the maximum pressure, depending on the set maximum pressure during boom lift and digging. When performing boom down, the two controllers start to limit the operator approximately 50 to 60 bar below the maximum pressure. Compared to the mechanical pressure feed reducer this means that the PID with bumpless transfer and the LQR with integral action limit the operator a lot, but when comparing with the physical system limitations mentioned in Section 7.2 it is not that bad. When closing the valve with a step signal, the valve has to be closed at least 60 bar below the maximum pressure during boom lift and 100 % operator signal and at least 20 bar below the maximum pressure during boom down and 100 % operator signal in order to not exceed the maximum pressure. However,

when closing the valve with a step the crane oscillates and causes vibrations in the backhoe. Thus, to be able to close the valve without oscillating the crane and cause vibrations in the backhoe the operator must be limited approximately at the pressure levels that the two controllers start to limit the operator.

8.2 Varying Dynamics

As mentioned in Chapter 7 different tunings were used for different cases, such as boom lift and digging. The reason for this was that the dynamics of the system varies significantly for the different cases, and thus different tunings were needed. This can clearly be seen in Section 7.4.3 where the tunings for the digging case were used for boom lift and vice versa. A good compromise was not possible to find since it would either overshoot for the digging case or limit the operator too much in the boom lift case. Therefore, another solution to the varying dynamics in the system is needed.

One way to solve the problem with varying dynamics could be to use gain scheduling, i.e. to use different controller parameters for different cases. The problem with this solution is that it would be necessary to identify the different scenarios in order to choose the correct controller parameters. This identification could possibly be achieved by using the pressure derivative, the difference in pressure in workport A and B, additional sensors such as cylinder position sensors, or any combination of these.

Some other ways to possibly handle the varying dynamics that were not investigated in this thesis are to use an adaptive controller that constantly adapts to the varying dynamics, or to use a more complete model that includes the crane on the backhoe. Including the crane on the backhoe in the model might help with the varying dynamics since they would be included in the model, for example movement of the cylinder would be included. The downside of this is, however, that the solution will be specific to the crane used, sensors for the cylinder positions would be needed and more data about the crane such as masses and inertias would be needed.

Not only are the dynamics of the system different for different scenarios, they can also vary from test to test in the same scenario. This can be seen in Figure 7.4 where both the PID controller and the LQRI controller reaches the maximum pressure, but it requires approximately 50 mA more for the LQRI controller compared to the PID controller to keep it at the maximum pressure. The reason for this is not that the controllers work differently, since they both have a reasonably constant control signal when this happens. Rather it is due to that the dynamics of the system changed between the tests, which could be explained by for example a change in oil temperature. This behaviour shows that the controllers used cannot be too sensitive to varying dynamics in order to work satisfactorily.

8.3 Oscillations & Notch Filter

When applying the controllers on a scenario where the cylinder moves, such as the boom lift test case, significant oscillations occurred when not using the notch filter described in Section 6.3. These oscillations can clearly be seen in Figure 7.9, where the PID controller was used without the notch filter to illustrate this. These oscillations likely occur due to the controllers exciting the system close to the resonance frequency of the crane on the backhoe, making it start to oscillate. The controllers do not dampen the oscillations since the natural frequency of the crane is not included in the model. Therefore, including the crane in the model could have reduced the oscillations, but then the model would have been dependent on the application, and cylinder position sensors would have been required.

The notch filter that was applied to the control signal efficiently reduced the oscillations, however it has some drawbacks. The theoretical drawbacks mentioned in Section 6.3 mean that the performance of the controllers can be degraded by the notch filter. Since it is only enabled after the peak of the pressure derivative it does not affect the initial closing of the valve, and thus does not increase the risk of exceeding the maximum pressure significantly. On the other hand, what can be seen in the figures in Section 7.3 is that after the initial closing the controllers generally have to increase the control signal again to reach the maximum pressure. This increase can be slightly delayed by the notch filter since it is enabled when this happens. Due to this the controller may limit the operator more than it needs to and can therefore make the machine feel unresponsive. This was however not noticed as a problem during testing.

Two additional changes could possibly improve the performance when using the notch filter. First, the resonance frequency of the machine could be identified online using for example recursive least squares. This would improve the filter since it would not have to be manually tuned, and it would likely make the frequency more accurate than a manually tuned one since it would adapt with varying dynamics. Having a more accurate frequency would make it possible to use a narrower notch filter, which would degrade the performance of the controllers less. Second, the notch filter could be included in the models used for controller synthesis. This would bring back the theoretical properties of the controllers, such as optimality, since the controller design would be aware of the filter. This was not investigated in this thesis since satisfactory results were obtained without it.

8.4 Dropping Pressure Phenomenon

As illustrated in Section 7.4.2, when the valve first opens the pressure drops approximately 10 bar. The most likely explanation to this is that the valve opens the port to the tank before the other port to the pump pressure. For example, when the valve is opened such that flow is directed from the pump to workport A, port B opens to tank slightly before A to the pump. This means that the pressure in workport B will drop before the flow is directed to port A. Since the valve is attached to a double acting cylinder, opening port B to the tank will also reduce the pressure in

port A since they are pushing the cylinder in opposite directions. Thus, a drop in the pressure can occur when opening the valve, and the controllers cannot do anything to mitigate this since it is a physical property of the system. There is however one way to avoid it, namely to never exceed the maximum pressure and thus never close the valve fully. This makes it even more important for the controllers not to overshoot the maximum pressure.

Another phenomenon that was noted in Section 7.4.2 was that after slightly opening the valve near the maximum pressure, the pressure slowly kept decreasing for about 2000 ms even though the controller slowly opened the valve. This is likely due to that the leakage in the valve is not overcome by the flow through the spool opening when the valve is almost closed. What is more interesting is that after 2000 ms the pressure suddenly rises. This phenomenon generally happens after the controller has closed the valve due to exceeding the maximum pressure and it is not specific to the PID controller. One possible explanation for the sudden pressure increase is that the spool area curve is very flat in the beginning as illustrated in Figure 2.2. This means that when the valve is almost closed, almost no flow is directed through the workport. It is not until the valve opens more that a significant amount of flow is directed through the workport. Therefore, the pressure can suddenly start to increase when the valve is opened enough to overcome the leakage in the valve. In order to verify this theory more information would be required, such as a sensor on the spool position in the valve.

If the sudden pressure increase is caused by the spool area curve, it could be solved by having a better approximation of the spool area curve in the controllers. This way the controllers could compensate for the flatter shape when the valve is almost closed. For example, three or more linear curves could be used depending on the current spool position. Alternatively, a nonlinear model could be used, which would also require a nonlinear controller. To apply any of these alternatives the spool position would have to be known, i.e. either a sensor for it would be required, or it would have to be estimated. Estimating it would likely not give a high enough accuracy due to manufacturing differences of pilot valves.

8.5 Model

The final model for control in (3.18) does not contain a model of the pressure compensator since the compensator instead was approximated with a constant pressure drop over the valve spool opening area, as mentioned in Section 3.1.4. In the results in Chapter 7 no phenomena that can be explained by the compensator approximation can be seen. This indicates that it was a suitable approximation to do when modelling the valve, and that it is unnecessary to model the pressure compensator when using the model of a directional control valve for pressure control in a software pressure feed reducer.

The linear approximation of the area curve seen in Figure 3.6 is quite inaccurate, especially when the valve is almost closed. As discussed in Section 8.4 a better approximation could be for example to use three or more linear functions for different spool positions, or a nonlinear function. This would not just solve the rising pressure phenomenon seen in Section 8.4. It would also make the controllers aware that the area curve is very flat when the valve is almost closed which would remove the creeping pressure behaviour that delays the time it takes to reach the maximum pressure both in simulations and reality.

9

Conclusion

The aim of the thesis was to evaluate different control algorithms used for pressure control when replacing the mechanical pressure feed reducer with a software solution, using the existing components in a directional control valve. The simulations and tests performed show that some of the control algorithms designed give a satisfactory behaviour for a pressure feed reducer. However, with the hardware used for implementation it was not possible to make the software pressure feed reducer perform as good as the mechanical one.

Some of the control algorithms developed in this thesis, such as the LQR with integral action, does not have to be applied as a software pressure feed reducer. Since the control algorithms themselves were developed for controlling pressure, and not specifically to limit it, they can also be used for general pressure control in directional control valves.

In Chapter 1 three research questions were formulated and throughout the thesis these questions have been answered. In the remainder of this chapter these questions will be briefly discussed, and the answers will be given. The first question to answer is:

- I. What is a suitable model of the electrohydraulic directional control valve when designing a software pressure feed reducer, and which parts of the system should be included in it?*

The model in (3.18) has, through both simulations and tests on a backhoe, shown to be accurate and detailed enough to use for the control algorithms developed in this thesis. This model does not include the dynamics of the pressure compensator, rather it assumes a constant pressure drop over the main spool. It also simplifies the load model to a constant volume of oil, which gave good results when combined with an integrator in the control algorithms. As discussed, a more detailed load model that includes the cylinder and crane could probably have improved the performance. However, it would require additional sensors and make the control algorithms more specific to the application.

Thus, to answer the first research question, a quite simple model of the directional control valve including only the dynamics of the main spool and a load as a constant volume appears to be suitable when designing a software pressure feed reducer.

The second research question to answer is:

II. Which model-based control algorithm is most suitable for a software pressure feed reducer in an electrohydraulic directional control valve when analysing oscillations, rise time, steady state error and overshoot compared to the mechanical pressure feed reducer?

Simulations and tests on the backhoe show that the two controllers that perform best are the LQR with integral action and PID with bumpless transfer. The two controllers with the added notch filter showed no major oscillations, had a reasonable rise time considering the system limitations and a steady state error of 4 bar in the worst case. In most cases no overshoot occurred, and when it did it was not major.

The LQR with integral action is clearly model-based, but it is not obvious that the PID controller with bumpless transfer is model-based since the controller parameters are not derived from the model. However, in this case it can be seen as model-based since the model is used to decide when the controller should be enabled. Thus, it is found to be one of the two model-based control algorithms that is most suitable for a software pressure feed reducer.

Compared to the mechanical pressure feed reducer the software solutions do not perform as well, even though they give a satisfactory behaviour in the backhoe. They start to override the operator at a significantly lower pressure than the mechanical solution due to limitations in the physical system. This leads into the last and third research question being:

III. Are there any limitations in the physical system when using model-based control algorithms for controlling the output pressure of an electrohydraulic valve?

The results from implementation of the control algorithms on the backhoe showed that it is not possible to make the software pressure feed reducer perform as good as the mechanical one. Especially it could be noted that in order to not exceed the maximum pressure, the software algorithms had to start limit the pressure 50 to 70 bar earlier than the mechanical pressure feed reducer during the boom lift and digging cases and 20 to 30 bar earlier during the boom down cases. This clearly shows that when using the spool in the directional control valve for limiting the output pressure the system dynamics are too slow. The pilot valve together with the spool have a too long time constant compared to the mechanical pressure feed reducer and this time constant limits the system when controlling the pressure. Despite this, it is possible to control the pressure, but with a limited speed.

Besides this, measuring only the pressure in workport A and workport B is limiting for the control algorithms performance. With additional inputs to the control system, like a spool position sensor and position sensors on the hydraulic cylinders the result could have been improved.

10

Future Work

Based on the results and the conclusion from this thesis some things that can be improved and further analysed have been found. Below a list of all these is presented.

- Further investigate what causes the \mathcal{H}_∞ Loop-Shaping controller to overshoot on workport A but not B, for example by testing shorter sampling times.
- Investigate if the MPC controller performs better with a longer prediction horizon and a higher number of solver iterations. This would require more computational power and could be tested by for example running the controller on a PC connected to the machine over CAN-bus.
- Use a better approximation of the spool opening area to make the pressure not creep towards the maximum pressure as slowly, and to possibly fix the sudden pressure increase when slowly opening the valve.
- Analyse further how well the LQR with integral action and PID with bumpless transfer controllers perform by for example
 - performing more tests on different scenarios.
 - having a more experienced and professional operator test the controllers for a longer time in practical application to evaluate the behaviour of the machine.
 - performing tests in a lab environment in order to measure performance under ideal conditions.
 - testing the controllers on different cranes and functions to evaluate the robustness of the algorithms.
- Make the control algorithms work for different scenarios without manually changing the tuning parameters. Two alternatives for this are to
 - implement gain scheduling for the already designed controllers and an algorithm that identifies the different scenarios.
 - implement an adaptive controller that works without gain scheduling.
- Implement an active damping algorithm in order to remove the last oscillations that can occur in some test scenarios.
- Investigate possible improvements when using spool position sensor and position sensors on the hydraulic cylinders attached to the crane.

10. Future Work

- Implement an algorithm to automatically find the notch frequency ω_o used in the notch filter.
- Investigate what can be changed in the directional control valve in order to make the dynamics faster and decrease the time constant of the system.

Bibliography

- [1] Parker Hannifin, *Technology in Motion: Mobile Hydraulic Technology*. Cleveland, USA: Parker Hannifin Corporation, 1999.
- [2] M. Axin, “Mobile working hydraulic system dynamics”, PhD thesis, Department of Management and Engineering, Linköping University, Linköping, Sweden, 2015.
- [3] S. Kim, J. Park, S. Kang, P. Y. Kim, and H. J. Kim, “A robust control approach for hydraulic excavators using μ -synthesis”, in *International Journal of Control, Automation and Systems*, vol. 16, 2018, pp. 1615–1628.
- [4] A. Makarow, J. Braun, C. Rösmann, G. Schoppel, I. Glowatzky, and T. Bertram, “Introduction of model predictive control for the system optimization of a proportional directional control valve”, in *IEEE Conference on Control Technology and Applications*, 2018, pp. 921–926.
- [5] M. Liermann, “Pid tuning rule for pressure control applications”, in *International Journal of Fluid Power*, vol. 14, 2013, pp. 7–15.
- [6] J. Boza, “Design and validation of an electro-hydraulic pressure-control valve and closed-loop controller”, PhD thesis, Department of Industrial and Mechanical Engineering, Western Michigan University, Michigan, USA, 2016.
- [7] *Instruktionsbok: Huddig 1160*, Utgåva 01-97, Hudiksvall, Sweden: Huddig, 1998.
- [8] *Sensors and switches for pressure, temperature, level and flow*, Cleveland, USA: Parker Hannifin Corporation, 2019. [Online]. Available: <https://www.parker.com/literature/HPCE/New/CAT-4083-UK.pdf>, Accessed on: Apr. 24, 2020.
- [9] *IQAN-LC6-X05 input devices: Electronic control systems*, Mölnlycke, Sweden: Parker Hannifin Electronic Controls Division, 2015. [Online]. Available: https://www.parker.com/Literature/Electronic%20Controls%20Division/Literature%20files/IQAN-LC6-X05_datasheet_HY33-8409-UK.pdf, Accessed on: Apr. 24, 2020.
- [10] *IQAN-MD4 display modules: Electronic control systems*, Mölnlycke, Sweden: Mobile Hydraulic Systems Division Europe, 2019. [Online]. Available: https://www.parker.com/Literature/Electronic%20Controls%20Division/Literature%20files/IQAN-MD4_datasheet_MSG17-8408-UK.pdf, Accessed on: Apr. 24, 2020.

- [11] *IQAN-MC4 master controller family: Electronic control systems*, Mölnlycke, Sweden: Mobile Hydraulic Systems Division Europe, 2017. [Online]. Available: https://www.parker.com/Literature/Electronic%20Controls%20Division/Literature%20files/IQAN-MC4x_brochure_HY33-8413-UK.pdf, Accessed on: Apr. 24, 2020.
- [12] *K170LS mobile directional control valve: Proportional, load sensing, pre-compensated*, Borås, Sweden: Parker Hannifin Mobile Controls Division Europe, 2010. [Online]. Available: https://www.parker.com/literature/Mobile%20Controls%20-%20Europe/HY17-8557-UK_K170.pdf, Accessed on: Apr. 24, 2020.
- [13] H. E. Merrit, *Hydraulic Control Systems*. New York, USA: John Wiley & Sons, 1967.
- [14] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. New York, USA: John Wiley & Sons, Inc., 1972.
- [15] D. S. Naidu, *Optimal Control Systems*. Boca Raton, USA: CRC Press, 2002.
- [16] K. J. Åström and R. M. Murray, *Feedback Systems: An introduction for Scientists and Engineers*. Princeton, USA: Princeton University Press, 2008.
- [17] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design and Tuning*. Research Triangle Park, USA: The Instrumentation, Systems, and Automation Society, 1995.
- [18] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. New York, USA: John Wiley & Sons, Inc., 2005.
- [19] K. Glover and J. C. Doyle, “State-space formulae for all stabilizing controllers that satisfy an H_∞ norm bound and relations to risk sensitivity”, in *Systems & Control Letters*, vol. 11, 1988, pp. 167–172.
- [20] J. C. Doyle and K. Glover, “State-space solutions to standard H_2 and H_∞ control problems”, in *IEEE Transactions on Automatic Control*, vol. 34, 1989, pp. 831–847.
- [21] D. J. Hoyle, R. A. Hyde, and D. J. N. Limebeer, “An H_∞ approach to two degree of freedom design”, in *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 2, 1991, pp. 1581–1585.
- [22] D. J. N. Limebeer, E. M. Kasenally, and J. D. Perkins, “On the design of robust two degree of freedom controllers”, in *Automatica*, vol. 29, 1993, pp. 157–168.
- [23] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. San Francisco, USA: Nob Hill Publishing, 2019. [Online]. Available: <https://sites.engineering.ucsb.edu/~jbraw/mpc/MPC-book-2nd-edition-2nd-printing.pdf>, Accessed on: Apr. 24, 2020.
- [24] A. Bemporad, N. L. Ricker, and M. Morari, *Model predictive control toolboxTM: User’s guide*, Natick, USA: Mathworks, 2019. [Online]. Available: https://www.mathworks.com/help/pdf_doc/mpc/mpc_ug.pdf, Accessed on: Apr. 24, 2020.

- [25] Y. Wang and S. Boyd, “Fast model predictive control using online optimization”, in *IEEE Transactions on Control Systems Technology*, vol. 18, 2008, pp. 267–278.
- [26] *IQANdesign version 6.02: User manual*, Mölnlycke, Sweden: Parker Hannifin Electronic Controls Division, 2020.
- [27] H. Zumbahlen, *Basic Liner Design*. Norwood, USA: Analog Devices Inc., 2007.
- [28] B. Mulgrew, P. Grant, and J. Thompson, *Digital Signal Processing: Concepts and Applications*. New York, USA: Palgrave MacMillan, 2003.
- [29] M. Mårlind, “Embedded implementation of active damping in hydraulic valves: An adaptive control solution”, Master’s thesis, Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden, 2017.

