Full-Bridge Oscillation Transformer Control

MASTER THESIS

Kristen Emil Bording Andersen
Electro-Mechanical System Design
Aalborg University
1. June 2023

AALBORG UNIVERSITY
STUDENT REPORT

Department of Materials and Production
Electro-Mechanical System Design
Fibigerstræde 14
9220 Aalborg Ø
https://www.mp.aau.dk/

Title:

Full-Bridge Oscillation Transformer Control

Project:

EMSD4 - Master Thesis

Project period:

February 2023 - June 2023

Author:

Kristen Emil Bording Andersen

Supervisor:

Per Johansen

Number of pages: 52
Number of pages in appendix: 35
Finished 01-06-2023

Preface

This project has been written in LaTeX. Sources are being referred to by the Harvard Method. Some sections are based upon methods of calculation from a single or multiple books. In these sections the book or books will be presented in the beginning of the section and a reference will not be added for every formula used. All figures and tables have been produced by the author unless otherwise stated. The nomenclature list includes numerical values of the constants used in the report with units switching between HYD and SI depending on ease of readability.

Kristen Emil Bording Andersen

Nomenclature

Supply Model

ω_P	Pump rotational velocity	rad/s
C_{dAcc}	Accumulator orifice discharge coefficient	0.7
D_P	Pump displacement	$12\,\mathrm{cm}^3/\mathrm{rev}$
$d_{OrificeAcc}$	Accumulator orifice diameter	10 mm
K_P	Pump controller proportional gain	
p_f	Accumulator fluid pressure	bar
p_{g0}	Accumulator initial gas pressure	30 bar
$p_{S,ref}$	Supply reference pressure	180 bar
p_S	Supply pressure	bar
Q_S	Pump flow	L/min
Q_{Acc}	Accumulator flow	L/min
Q_{Si}	*for* i *= 1,2* Supply flow i to FBoT	L/min
V_f	Accumulator fluid volume	L
V_g	Accumulator gas volume	L
V_S	Volume of supply line	1 L
V_{f0}	Accumulator initial fluid volume	2 L
V_{g0}	Accumulator initial gas volume	2 L

Fluid Basics

$\eta_{orifice}$	Orifice Efficiency	bar
k_v	Orifice Constant	$\frac{\mathrm{m}^3}{\mathrm{s}\sqrt{\frac{\mathrm{N}}{\mathrm{m}^2}}}$
p	Pressure	bar
Q	Flow	m^3/s
V	Volume	m^3
x_v	Orifice Opening	%

Load System

A_A	Load cylinder piston area A	$23.76\,\mathrm{cm}^2$
A_B	Load cylinder piston area B	$11.19\,\mathrm{cm}^2$
B_{cyl}	Load cylinder damping coefficient	100 Ns/m
F_{load}	Load force	N
m_c	Load mass	289.8 kg
V_{dCA}	Dead volume load cylinder chamber A	1 L

Control System

α	Input-Output pressure ratio	
$\dot{\boldsymbol{q}}$	Time Differentiated System States	
$\boldsymbol{A}_{ineq}$	Linear Inequality Constraint Matrix	
$\boldsymbol{b}_{ineq}$	Linear Inequality Constraint Vector	
$\boldsymbol{b}_l$	Design Variable Lower Bounds	s
$\boldsymbol{b}_u$	Design Variable Upper Bounds	s
$\boldsymbol{c}_{grad}$	Constraint Gradient Matrix	
$\boldsymbol{c}_{ineq}$	Nonlinear Inequality Constraints	
$\boldsymbol{q}$	State Vector	
$\boldsymbol{Q}_{error}$	Positional Constraint Violation Vector	
$\boldsymbol{t}_{B,HP,[off,on]}$	Row Vector of Switching Times For Low Pressure Valve Inputs For the Braking Role	s
$\boldsymbol{t}_{B,HP,[on,off]}$	Row Vector of Switching Times For High Pressure Valve Inputs For the Braking Role	s
$\boldsymbol{t}_{des,id}$	Vector of Design Variable Switching Time Role Identification	
$\boldsymbol{t}_{des}$	Design Variable Vector	s
$\boldsymbol{t}_{I,HP,[off,on]}$	Row Vector of Switching Times For Low Pressure Valve Inputs For the Input Role	s
$\boldsymbol{t}_{I,HP,[on,off]}$	Row Vector of Switching Times For High Pressure Valve Inputs For the Input Role	s
$\boldsymbol{t}_{O,HP,[off,on]}$	Row Vector of Switching Times For High Pressure Valve Inputs For the Output Role	s
$\boldsymbol{U}$	Input Matrix	
$\boldsymbol{u}$	Input Vector	

$\boldsymbol{v}_{p,Bidx}$	Vector of Time Steps Where The Braking Pressure is High	
$\boldsymbol{y}$	Output Vector	
β_m	Bulk Modulus for Prediction Model	Pa
δW_k	Partial Work at Time Step k	J
Δx_p	Change in Piston Position	m
$\dot{x}_{p,k}$	Piston Speed at Time Step k	m
η	Efficiency of Outputs Pressure Work on Piston Compared to Input Pressure Work on Piston	
B	Viscous Damping Coefficient	Ns/m
B	Viscous Damping Coefficient	m
c_B	Braking Pressure Timing Constraint Violation	
c_Q	Positional Constraint Violation Sum	
c_1	Objective Function Gradient Entry for First Design Variable	
c_2	Objective Function Gradient Entry for Second Design Variable	
c_{Bg}	Braking Pressure Timing Constraint Gradient	
F	Force from pressure on piston	N
J	Cost Function	
k	Time step	
k_p	Number of Time Steps in Prediction Horizon	
k_{peak}	Time Step at Piston Peak Position	
p_1	Chamber 1 Pressure	Pa
p_2	Chamber 2 Pressure	Pa
p_3	Chamber 3 Pressure	Pa
p_4	Chamber 4 Pressure	Pa
p_B	Braking Pressure	Pa
p_I	Input Pressure	Pa
p_O	Output Pressure	Pa
$p_{B,HP}$	Manifold High Pressure Value for Braking Chamber	Pa
$p_{I,HP}$	Manifold High Pressure Value for Input Chamber	Pa
$p_{O,HP}$	Manifold High Pressure Value for Output Chamber	Pa

$Q_{error,i}$	Entry at Time Step i of Positional Constraint Violation Vector	
$Q_{i,cv,in}$	Prediction Model Inflow to Chambers $i = 1$ to 4 via Check Valves	m^3/s
$Q_{i,cv,out}$	Prediction Model Outflow from Chambers $i = 1$ to 4 via Check Valves	m^3/s
$Q_{i,in}$	Prediction Model Inflow to Chambers $i = 1$ to 4 via On/Off Valves	m^3/s
$Q_{i,out}$	Prediction Model Outflow from Chambers $i = 1$ to 4 via On/Off Valves	m^3/s
t_1	Switching Time to turn on u_1	s
t_2	Switching Time to turn off u_1	s
t_3	Switching Time to turn on u_2	s
t_4	Switching Time to turn off u_2	s
t_5	Switching Time to turn on u_3	s
t_6	Switching Time to turn off u_3	s
t_7	Switching Time to turn on u_4	s
t_8	Switching Time to turn off u_4	s
t_9	Switching Time to turn off u_5	s
T_p	Time at End of Prediction Horizon	s
T_s	Controller Sample Time	0.1 ms
t_{10}	Switching Time to turn on u_5	s
t_{11}	Switching Time to turn off u_6	s
t_{12}	Switching Time to turn on u_6	s
t_{13}	Switching Time to turn off u_7	s
t_{14}	Switching Time to turn on u_7	s
t_{15}	Switching Time to turn off u_8	s
t_{16}	Switching Time to turn on u_8	s
$t_{des,1}$	First Design Variable, Describing Input HP Turn Off Time	s
$t_{des,2}$	Second Design Variable, Describing Braking HP Turn On Time	s
T_{end}	End of Plant Simulation	s
t_{lag}	Time Lag to Simulate Valve Delay	2 ms
t_{pert}	Perturbation Time for Objective Function Gradient Analysis	s
T_{ps}	Prediction Horizon Sample Time	s
u_1	Valve Input Signal for Supply Chamber 1 High Pressure	

u_2	Valve Input Signal for Supply Chamber 2 High Pressure	
u_3	Valve Input Signal for Load Chamber 3 High Pressure	
u_4	Valve Input Signal for Load Chamber 4 High Pressure	
u_5	Valve Input Signal for Supply Chamber 1 Low Pressure	
u_6	Valve Input Signal for Supply Chamber 2 Low Pressure	
u_7	Valve Input Signal for Load Chamber 3 Low Pressure	
u_8	Valve Input Signal for Load Chamber 4 Low Pressure	
u_B	Valve Input Signal for Chamber With Braking Role	
u_I	Valve Input Signal for Chamber With Input Role	
u_O	Valve Input Signal for Chamber With Output Role	
$v_{p,Bidx,i}$	First Entry of Vector of Time Steps Where The Braking Pressure is High	
W	Work	J
W_I	Input Work	J
W_O	Output Work	J
W_{loss}	Work Lost	J
$x_{p,k}$	Piston Position at Time Step k	m
x_{ref}	Piston Position Reference	m
z_i	Delayed Valve Input Signal for Prediction Model, On/Off Valves $i = 1$ to 8	

Fluid Constants

α_μ	Pressure-viscosity coefficient	0.02×10^{-8}
α_{air}	Percentage air in hydraulic oil	2 %
β_0	Asymptotic Bulk modulus	16 000 bar
β_{eff}	Effective bulk modulus	bar
μ	Dynamic viscosity of oil	Pa · s
μ_0	Dynamic viscosity of oil at atmospheric pressure	40.02 mPa · s
ρ_f	Hydraulic oil density	866 kg/m^3
n	Polytropic constant adiabatic process	1.4

Piston Parameters

$\ddot{x}_p$	Piston acceleration	m/s^2
$\dot{x}_p$	Piston velocity	m/s

τ_s	Shear stress	MPa
A_s	Shear stress Area	m^2
A_{set}	Areas on each piston face, that the pressure works on	61 mm^2
d_i	*for i = 1..3* Piston Diameters	10.0 mm,13.3 mm,16.0 mm
F_p	Force due to pressures on piston	N
F_s	Shear force	N
h	Gap distance between piston and chamber	25 $^-$m
L_i	*for i = 1..3* Piston lengths	12.5 mm,40.0 mm,40.0 mm
L_s	Piston max travel length	35 mm
m_p	Piston mass	1.519 kg
p_{Ci}	*for i = 1..4* Chamber pressures	bar
$Q_{le,(from,to)}$	Leakage flows from chamber to chamber or chamber to tank	L/min
V_{dC}	Dead volume in each chamber	25.9 mL
x_p	Piston position	m

Valve Model

τ_{Vd}	Valve de-energizing time constant	0.0050
τ_{Ve}	Valve energizing time constant	0.0033
G_V	Valve transfer function	
p_{CV}	Minimum pressure difference check valve	0.5 bar
$p_{Nom,CV}$	Check Valve nominal pressure	5 bar
$Q_{Nom,CV}$	Check Valve nominal flow	30 L/min

Table of Contents

Abstract

This project focuses on the development of a control system for a Full Bridge Oscillation Transformer (FBoT) to efficiently transform fluid power. The FBoT, which is has been developed at Aalborg University utilizes a mechanically simple system consisting of a free-floating piston that in connection with its housing forms four pressure chambers. It's basic working principle centers on converting fluid power into kinetic energy and then back into fluid power at a different pressure. A detailed model of the system is developed and validated using lab data. Model Predictive Control (MPC) is chosen as the control strategy, specially designed for the FBoT's unique characteristics, where valve switching times are optimized for a prediction horizon. The control system successfully enables reciprocal piston motion, power transformation, and robust operation in different modes, which is demonstrated on a virtual plant. For implementation on a lab setup, further development is needed, which primarily concerns the deployment of an observer to estimate fluid viscosity.

Chapter 1
Introduction

In the hydraulic equipment industry, there are put great efforts into the pursuit of improving efficiency, which in turn benefits the reduction of CO_2 emissions, much in accordance with the extensive ambitions of various industries and entire nations. Progress is achieved through electrification and energy recovery strategies among other things (Volvo CE, 2021). A component which is a culprit of the inefficiencies in hydraulics is the use of the proportional valve, which popularly can be compared to controlling the speed of car with the brakes while applying constant full throttle. It is therefore of great interest to explore alternatives, which happens at companies, research facilities and universities, including Aalborg University (AAU). Here, there have been invented, designed and constructed a linear digital hydraulic transformer which is aimed to replace the encumbering proportional valve, proved by installing it in an existing setting in an electric wheel loader seen in figure 1.1, located at AAU.

The transformer is called a Full Bridge Oscillation Transformer (FBoT), which is a simple mechanical mechanism, actuated only by on-off valves. The FBoT does however require a robust control system which is the focus of this report. The initial problem statement is therefore:

"How can the FBoT control system be designed to be robust while maximizing the transformation efficiency?"

(a) VM Loader LXE 1026 electric wheel loader ***(b)*** Installation of the FBoT in the wheel loader

Figure 1.1. Wheel Loader from Vinderup Maskiner (2023)

Chapter 2
System Description

This chapter serves to describe the FBoT system and to elaborate its driving ideas.

2.1 Fundamental Concept

The fundamental concept of the FBoT is to transform fluid power as efficiently as possible. This is in contrast to the conventional proportional valve drives which inherently are inefficient. Conventional proportional valves can be modelled as orifices, where the flow is altered proportionally to the valve opening for some pressure difference. The general orifice equation can be written in a simple form as equation 2.1. (Hansen, 2023).

$$Q = x_v k_v \sqrt{|p_A - p_B|} \tag{2.1}$$

where k_v is a constant containing discharge coefficient, opening diameter and density terms, x_v describes how open the orifice is with a value between zero and 1, and p_A and p_B are pressures on either side of the orifice. The losses that are inherent to this flow restriction can be shown by considering fluid power pQ. The flow on either side of an orifice is equal, which means that the efficiency of an orifice is proportional to the pressure difference as:

$$\eta_{orifice} = \frac{p_B Q}{p_A Q} \tag{2.2}$$

where the flow Q cancels out. It is also apparent from the orifice equation that to increase flow, the pressure difference must be increased which in turn decreases efficiency. This relationship can be shown on a pressure-flow diagram as in figure 2.1. In the same figure, the concept of equal power transformation is introduced. The curves shown are constant power lines i.e. $pQ = const. \Rightarrow p_A Q_A = p_B Q_B$. The orifice jumps from one iso-power-line to the next, but to stay on the same line would in theory be a loss-free transformation. This is the driving idea of the FBoT: To maintain energy energy while controlling flow. Where it is quite straight forward

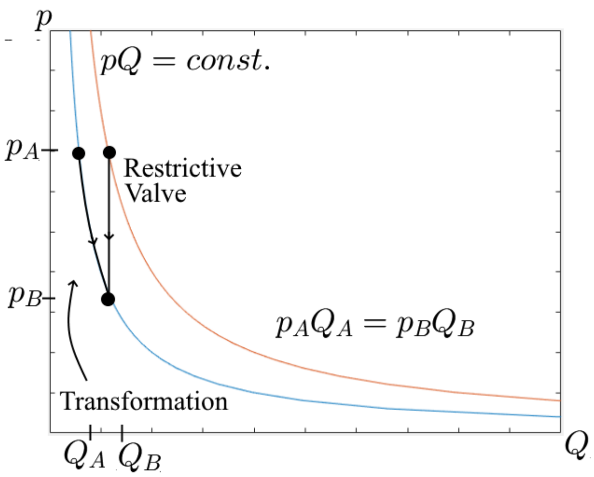


Figure 2.1. Pressure Flow Diagram of proportional/restrictive valve vs. transformation

to control flow for some pressure difference with a proportional valve, it will be more complicated using the transformation device which will be elaborated.

2.2 FBoT Operating Principles

The FBoT works by controlling the pressures surrounding a free-floating piston in a housing by manipulating on/off valves. The FBoT is able to transform power from a supply line to a hydraulic load and vice versa, which makes it possible to regenerate energy. The system assumes a common pressure rail (CPR) including accumulators, since an inherent feature of the FBoT is to circulate supply line flow. The FBoT consists of four chambers, two of which are connected to the supply manifold, and two connected to the load manifold. All chambers are meanwhile also connected to the tank. All connections are controlled with on/off valves alongside check valves which gives a total of four valve connections per chamber. These properties makes it possible to control the FBoT in numerous ways, which will be elaborated upon later. The main idea is to let high pressure (HP) fluid into one of the FBoT chambers, and let the energy be converted to kinetic energy in the piston mass. This kinetic energy is then used to compress fluid in a chamber on the opposite side of the piston. When the pressure matches that of the manifold, fluid flows out of the chamber, thereby succeeding a transformation.

2.2.1 Transformation Cycles

The FBoT is capable of several kinds of transformations which can be put in two categories; pumping mode and regeneration mode. When in pumping mode, the goal of the FBoT is to transform power from the supply manifold to the load manifold. Figure 2.2 shows 12 steps of a transformation cycle, where the valves, check valves and connections have been simplified. Each chamber can be connected to either tank “T” or respective manifolds, “S” for supply and “L” for load. Colored arrows indicate open valves and the arrow on the piston indicates moving direction. The colors in the chambers symbolize pressure levels; blue is tank pressure, red is supply pressure, orange is load pressure and green is an intermediate pressure. A whole cycle is shown, which means the piston returns to its starting position. It will however be evident that the transformation could be shown for only a half-cycle, since two half-cycles are very similar. This is because both sides are symmetrical and both supply chambers are connected to the same manifold and likewise for the load chambers.

1. The right-side supply port is open, and all others are closed, except for tank access for the load chambers and the left side supply chamber. High pressure fluid enters the chamber which creates a force on the piston that accelerates it.
2. The left-side load chamber is disconnected from both manifold and tank, and since the piston is compressing the chamber, the fluid will pressurize.
3. When the load chamber has been sufficiently compressed, its pressure will match and quickly surpass that of the load manifold, and fluid will flow through an open valve into the load manifold.
4. The piston must come to a stop before reaching the left side of the housing, and the right-side supply port is therefore closed off. This creates a decompression in the chamber, since the piston still is moving. Meanwhile, the left-side supply port, which until now has been connected to tank is also closed off. This creates a compression in the left-side supply chamber. The piston will begin to decelerate.
5. The piston is still moving and will compress the left-side supply chamber such that the chamber pressure eventually matches that of the supply manifold. Since the piston is still moving to the left, there will be let pressurized fluid back into the supply manifold.
6. The piston reaches a standstill momentarily where no fluid is displaced. The transformation process has completed a half-cycle where energy from the supply has been converted to

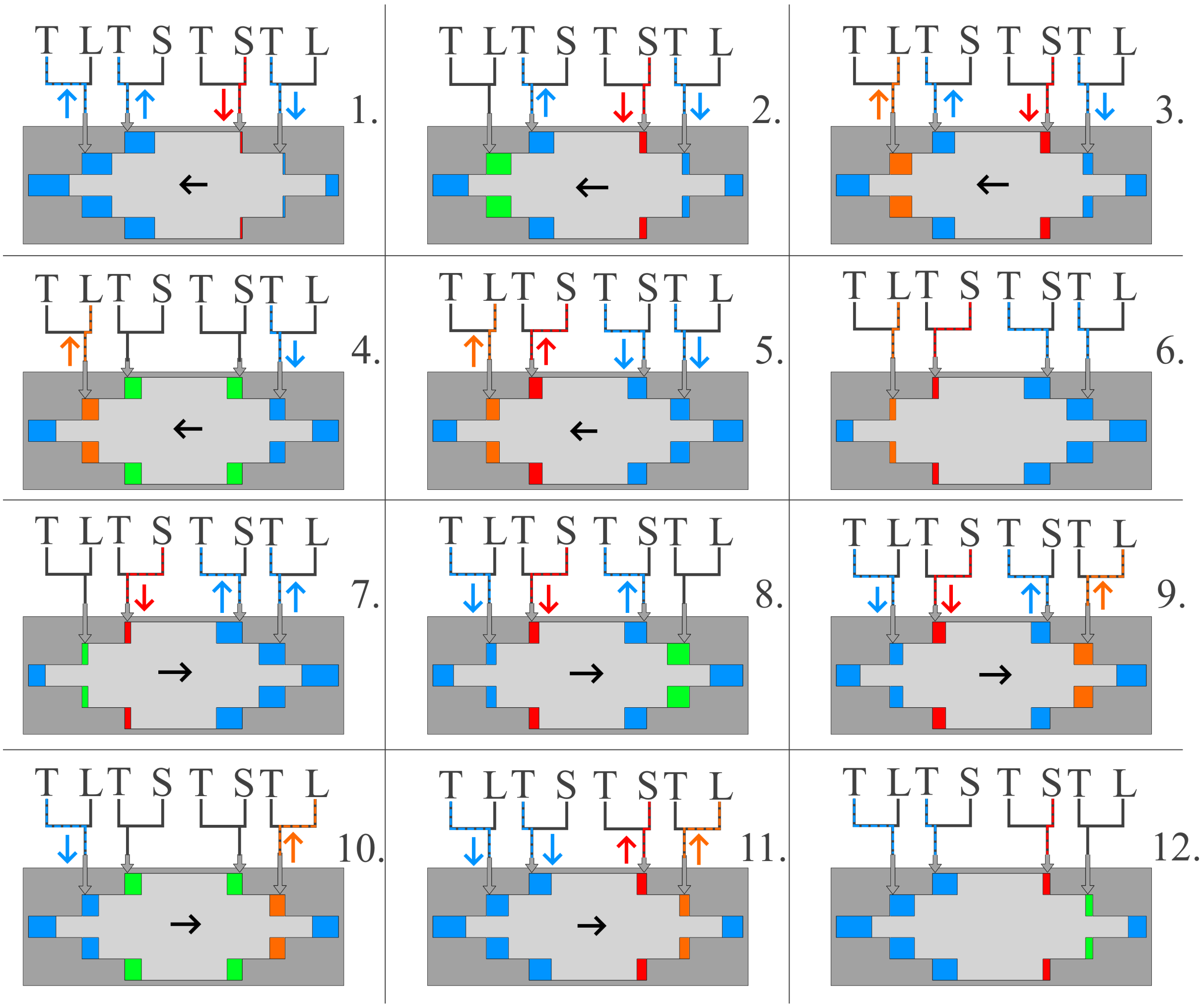


Figure 2.2. Transformation Diagram Supply to Load

kinetic energy in the piston and back into fluid on the load side and a little amount back into the supply, and some energy has been lost due to friction.

7. At this point, the next half-cycle can begin. The left-side supply port is open, and the left-side load port is still closed. The piston accelerates, which decompresses the left-side load chamber. The rest of the steps *8* through *12* describe the back-stroke which in all regards are similar to the forward stroke. At step *12* the whole cycle is complete and a new cycle can begin.

The FBoT is also capable of energy regeneration i.e. transforming power from the load manifold to the supply side. The transformation diagram of load to supply would be exactly like the one shown for supply to load side, only that the load chambers would be used to accelerate the piston, and the supply chambers would absorb the kinetic energy and create outflow. It is also possible to have an idling cycle, where no transformations occur. In that case, the output chambers are connected to the tank, and the input chambers are used to accelerate and decelerate the piston. An idling cycle is however only theoretically loss-free since there will be energy losses due to viscous damping. Energy-wise, it will therefore be most sound to avoid idling cycles.

2.3 Hydraulic Diagram

Where the previous section served to show the operational principles of the FBoT, this serves to show the details of the hydraulic diagram. The FBoT is installed in an electric wheel loader as described in the introduction, and the diagram takes basis in that configuration with some simplifications regarding the boom and hydraulic pump. The hydraulic diagram is showed in figure 2.3.

In this setup, both load chambers are connected to the same load manifold, which is connected to the load cylinder. A variable speed pump is connected to the supply manifold. For the supply side, there is an accumulator which will be part of the subsequent modelling, but the at load side, there is also an accumulator, which however not will be modelled for the scope of this report.

It can be seen from the diagram, that there are 8 on/off valves and 8 check valves. The check valves ensure that chamber pressure never will become larger than manifold pressures, and can be useful in fluid transformation since they do not need control signals to function, and when a chamber pressure surpasses a manifold pressure, it is often desired to let the fluid out into the manifold. They also ensure that fluid from tank flows into chambers experiencing decompression.

The free floating piston is completely enclosed in a housing, and its varying cylindrical geometry combined with the housing geometry creates four main chambers C1, C2, C3 and C4 that can be pressurized. There are also three other chambers which are permanently connected to tank and will therefore not have an effect in terms of forces on the piston. The piston is symmetrical and the piston surface areas in each chamber perpendicular to the moving direction are equal, so equal opposing pressures will create a zero net force. The piston is sliding in an oil film and there will be viscous damping during movement, and since there is no perfect sealing between chambers, there will be some leakage cross chambers.

Continuity equations and further description of flows are found in appendix A.

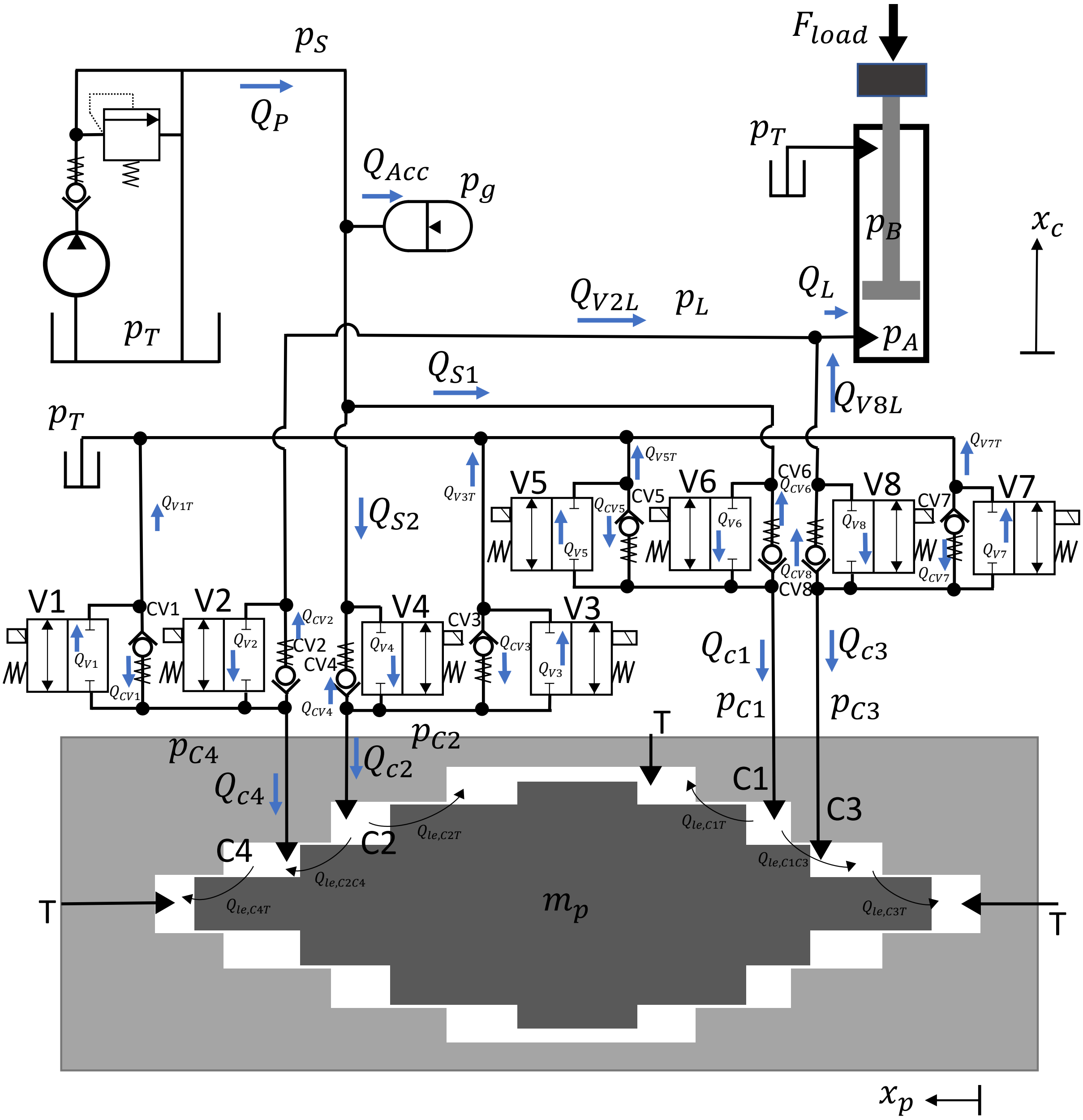


Figure 2.3. Hydraulic diagram of FBoT

Chapter 3
Modelling

The purpose of this chapter is to develop a time domain dynamic model of the hydraulic system which can be used for control system development. That is a model where the on/off valves function as inputs, resulting in pressure changes, flow and mechanical movement. The hydraulic system consists of numerous parts, and this chapter will elaborate on each important feature. The fluid power fundamentals book by Hansen (2023) is used extensively as a source for modelling hydraulic components. This primarily includes the general orifice equations, continuity equations, bulk modulus models and viscosity models. Other sources are cited when used. Throughout the chapter, various constants are used, and their numerical values can be found in the nomenclature section. The models presented throughout this chapter are implemented in MATLAB's Simulink environment, and an overview of the Simulink model is found in appendix B.

3.1 Fluid Models

Throughout the subsequent parts of the model, some fluid properties will appear multiple times. These are models of the stiffness of the hydraulic oil, bulk modulus, and the viscosity of the oil.

3.1.1 Bulk Modulus

Bulk modulus is a parameter describing the stiffness of the hydraulic oil, which is dependent on the pressure of the oil and air content. This parameter is important since it is part of the continuity equation. The bulk modulus of the hydraulic oil is modelled as a fluid-air mixture:

$$\beta_{eff}(p) = \frac{(1-\alpha_{air})\exp\left(\frac{p_0-p}{\beta_0}\right) + \alpha_{air}\left(\frac{p_0}{p}\right)^{\frac{1}{n}}}{\frac{1-\alpha_{air}}{\beta_0}\exp\left(\frac{p_0-p}{\beta_0}\right) + \frac{\alpha_{air}}{np_0}\left(\frac{p_0}{p}\right)^{\frac{n+1}{n}}} \tag{3.1}$$

Where β_0 is the asymptotic bulk modulus value as the pressure goes towards infinity, $\beta_{eff}(p)$ is the effective bulk modulus as a function of pressure, α_{air} is the percentage of air in the oil, n is a polytropic constant for adiabatic processes and p_0 is the atmospheric pressure. Bulk modulus as a function of pressure is showed in figure 3.1. It is seen that the stiffness varies significantly and non linearly due to pressure. At low pressures, a pressure change results in a large change of stiffness, but at higher pressures, the resulting change is smaller, and the stiffness plateaus at β_0. Since chambers pressure will vary from tank levels to supply and load levels, the non-linearity of the oil stiffness can have a pronounced effect.

3.1.2 Viscosity

The viscosity of the oil is primarily used for modelling the leakage flows and the viscous shear forces. There exist various viscosity models, and the one that will be used here is the Barus equation which is a pressure dependent viscosity model (Knežević and Savić, 2006).

$$\mu(p) = \mu_0 e^{\alpha_\mu p} \tag{3.2}$$

Where $\mu(p)$ is the dynamic viscosity as function of pressure, μ_0 is the dynamic viscosity at atmospheric pressure, α_μ is a pressure-viscosity coefficient depending on pressure and

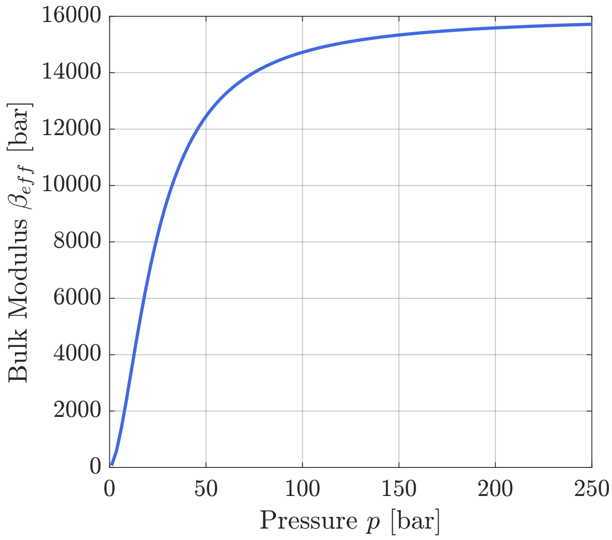


Figure 3.1. Effective Bulk modulus at different pressures

temperature, and p is pressure. Temperature is an unknown factor in both the model, and on the lab setup, so this will not be included. Instead, $\alpha_\mu = 0.02e-8$ is locked to an initial value which can be tuned during model validation. Figure 3.2 shows how the viscosity changes depending on the pressure. It is seen that the viscosity increases for an increasing pressure, and while the effect is small it will presumably make the model more accurate.

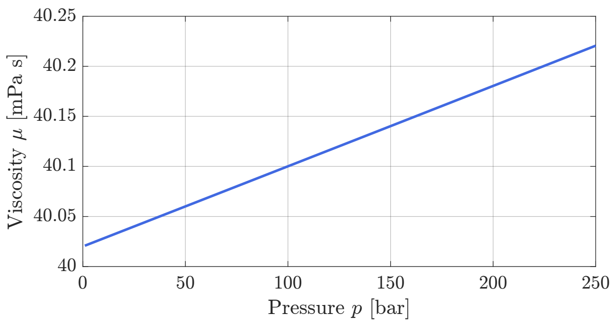


Figure 3.2. Barus viscosity model at different pressures

3.2 Pump and Supply

The pump and supply system is modelled as three parts: The pump which creates a supply flow, a continuity equation for the supply pressure, and lastly the accumulator.

Supply Flow

The pump motor speed is variable, and this is used to control the supply flow Q_S. A reference supply pressure $p_{S,ref}$ is defined and compared to the actual supply pressure p_S. The difference is used to drive the displacement pump to create the flow i.e. a proportional flow controller.

The displacement pump creates a flow as

$$Q_P = D_P \omega_P \tag{3.3}$$

The control law for the motor speed is

$$\omega_P = (p_{S,ref} - p_S) K_P \tag{3.4}$$

The proportional gain K_P is a tuning parameter, and a saturation block ensures that the pump operates within its specifications.

Accumulator

The purpose of an accumulator is ordinarily to reduce pressure oscillations, but for the FBoT it does more, as it is an inherent property that high pressure fluid flow back into the supply line from the transformer, and the accumulator captures this fluid. The bladder accumulator is modelled using the ideal gas equation with a polytropic constant $n = 1.4$ used for adiabatic processes.

$$pV^n = const \tag{3.5}$$

The bladder has an initial volume of $V_{g0} = 2\,\mathrm{L}$, and the precharge pressure is p_{g0}, subscript "g" meaning gas. The fluid pressure p_f in the accumulator must be equal to the gas pressure since the membrane has an insignificant inertia. The total volume inside the accumulator must be constant, so the gas volume V_g can be determined as

$$V_g + V_f = V_{g0} + V_{f0} \Rightarrow V_g = V_{f0} + V_{g0} - V_f \tag{3.6}$$

The pressure of the fluid in the accumulator can then be found using the gas equation as

$$p_f = \frac{p_{g0} V_{g0}^n}{V_g^n} \tag{3.7}$$

Figure 3.3 shows a plot of the inverse relationship between gas volume and pressure

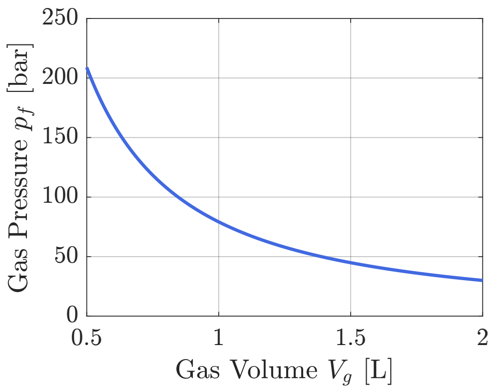


Figure 3.3. Accumulator gas/volume relation

The flow into the accumulator is modelled using the orifice equation as

$$Q_{Acc} = C_{dAcc} \frac{d_{OrificeAcc}^2}{4} \pi \sqrt{\frac{2}{\rho_f}} \sqrt{|p_S - p_f|}\ \mathrm{sign}(p_S - p_f) \tag{3.8}$$

The volume of the fluid inside the accumulator can then be found by integrating the flow

$$V_f = \int Q_{Acc} dt + V_{f0} \tag{3.9}$$

3.2.1 Pressure in Supply Line

The pressure in the supply line is modelled using the continuity equation, that generally can be stated as:

$$Q_{in} - Q_{out} = \dot{V} + \frac{V}{\beta}\dot{p} \tag{3.10}$$

Applying this to the supply line, the continuity equation can be reformulated as:

$$\dot{p_S} = \frac{\beta_{eff}}{V_S}(Q_P - Q_{S1} - Q_{S2} - Q_{Acc}) \tag{3.11}$$

The flows Q_{S1} and Q_{S2} are the supply lines going into the FBoT. The supply pressure p_S is found by integration. The pressure is limited to 250 bar to simulate the pressure relief valve. The subsystem of the supply system is simulated in Simulink. The pressure reference is set at $p_{S,ref} = 180$ bar, the initial supply pressure is $p_{S,ini} = 2$ bar, and the out-flow of the system is set to 10 L/min. The time data is showed in figure 3.4.

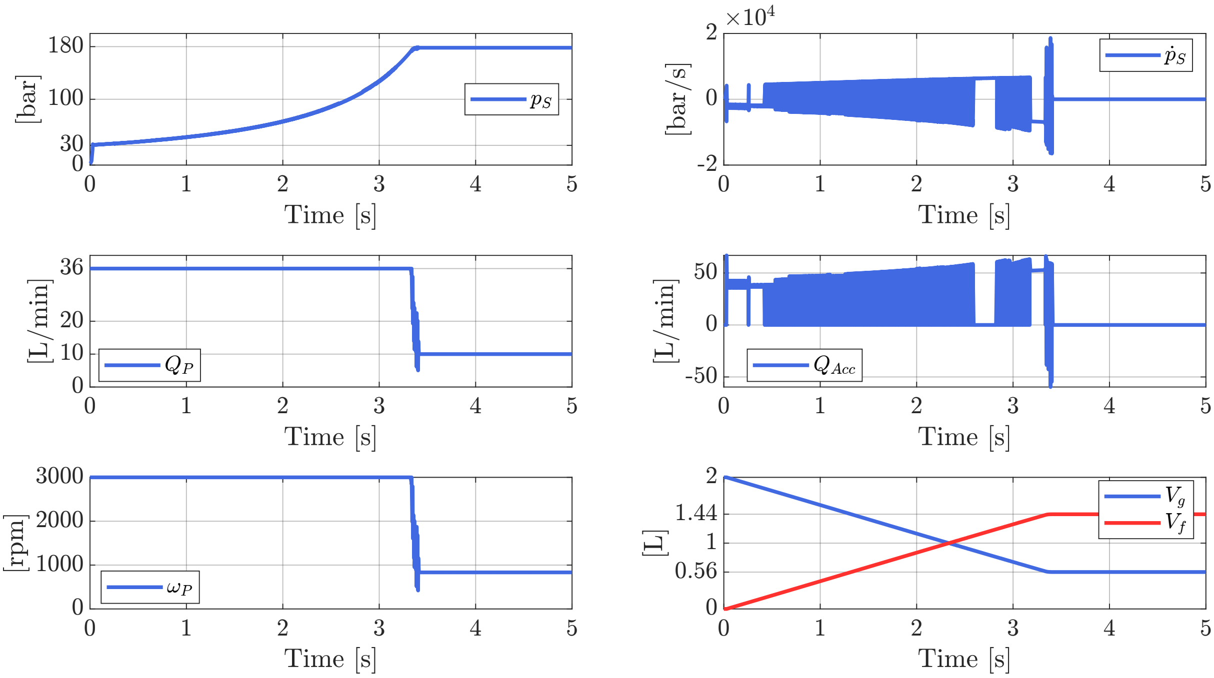


Figure 3.4. Data from test of supply system model

It is seen that the supply pressure starts at the initial value and very quickly rises to 30 bar which is the pre-pressure of the accumulator. At this point, oil starts flowing into the accumulator, and the supply pressure rises exponentially, which corresponds to the relationship between pressure and volume in the accumulator. The pressure reaches a steady state at the supply pressure reference. When the supply pressure reaches the reference, the pump velocity becomes lower and reaches a steady state where the supply flow becomes equal to the out-flow at 10 L/min. The accumulator flow is positive until the supply pressure reaches the reference pressure, and at that point, the accumulator flow shortly fluctuates and quickly becomes zero. This complies with the gas volume V_g and fluid volume in the accumulator V_f being constant along with the supply pressure. If the supply pressure were to fall, then the accumulator flow would become negative (flow out of the accumulator) until the gas pressure and supply pressure becomes equal. The steady state volumes in the accumulator can be used as initial values in pair with the supply pressure steady state value as initial values.

The simulation seems to make sense and this concludes the part of the model concerning the supply line. The pump control tries to maintain a reference supply pressure by delivering a pump flow to the system, while the accumulator can absorb surplus flow going into the supply system from the FBoT.

3.3 Valves

There are 16 valves connected to the FBoT chambers, where 8 of them are check valves and the other 8 are on/off valves. One of each will be modelled in this section, since the only difference between the models are the different up/down stream pressures and flows which can be seen from the hydraulic diagram fig. 2.3 on page 6.

3.3.1 Check Valves

The check valves only permit flow in one direction and are modelled by the orifice equation. The spring dictates a minimum pressure difference p_{CV} before flow occurs. Figure 3.5 illustrates a check valve.

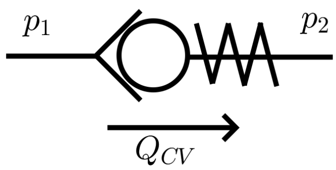


Figure 3.5. Check Valve Component

The flow is modelled by the orifice equation

$$Q_{CV} = \begin{cases} \frac{Q_{Nom,CV}}{\sqrt{p_{Nom,CV}}} \sqrt{|p_1 - p_2|} & (p_1 - p_2) > p_{CV} \\ 0 & \text{All other cases} \end{cases} \tag{3.12}$$

Figure 3.6 shows a simulation of a check valve subjected to a varying and constant pressure. It is seen that only when p_1 is larger than p_2 there is flow through the valve, which shows that the check valve functions as intended.

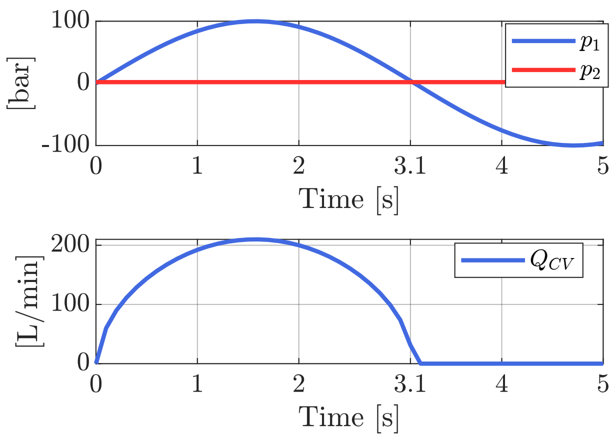


Figure 3.6. Check Valve Test

3.3.2 On/Off Valves Flow

The on/off valves permits flow in either direction, but naturally only if it is open. An on/off valve is illustrated in figure 3.7.

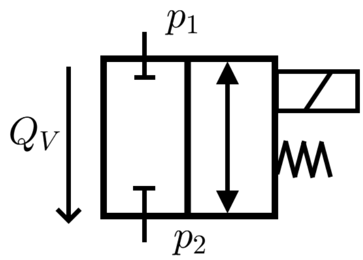


Figure 3.7. On/Off Valve Component

The flow is modelled using the orifice equation as:

$$Q_V = \frac{Q_{Nom,CV}}{\sqrt{p_{Nom,CV}}} z_V \sqrt{|p_1 - p_2|}\ \mathrm{sign}(p_1 - p_2) \tag{3.13}$$

The On/Off action is achieved by the variable term z_V. This determines how open the valve is by a value between 0 and 1. In an ideal world, the degree of openness would be either open or closed, but the valve will exhibit some dynamic behaviour which will be modelled as well.

3.3.3 On/Off Valves Dynamics

The On/Off valves receive a control signal u_V, which results in the solenoid in the valve moving which opens the valve z_V. The used valve is a WS22GD5400P from Bucher Hydraulics and the data sheet is found in appendix C.1. Dynamic response is not given in the data sheet, but switching times are stated as 6 ms to 20 ms for energizing, and 10 ms to 30 ms for de-energizing, depending on flow rate, pressure, viscosity and dwell time according to the data sheet.

A valve is a dynamic system with moving parts, and will be modelled as a first order dynamic system, but instead of developing a mechanical model, the valve model parameters will be tuned to match the specifications of the data sheet at first, and can later be tuned according to experimental data if required. The system will be formulated as:

$$\dot{z}_V = -Az_V + Bu_V \tag{3.14}$$

This is a state space equation, but can also be formulated in the laplace domain as a transfer function as:

$$\frac{z_V(s)}{u_V(s)} = \frac{B/A}{\frac{1}{A}s + 1} \tag{3.15}$$

From this standard formulation, the time constant of the system is $\tau = 1/A$ and the gain is B/A, and thus the parameters can initially be found. A time delay of T_d will be included, and with a desired settling time of $T_{s,v}$, and assuming a settling time of 4 time constants, the A parameter be calculated as:

$$A = \frac{4}{T_{s,v} - T_d} \tag{3.16}$$

The steady state value of z_V should be unity, so $B = A$. Since the settling times should be different depending on whether the input is energizing or de-energizing, a simple if statement is included in the model to toggle between the model parameters. Choosing $T_{s,energizing} = 20\,\mathrm{ms}$ and $T_{s,de-energizing} = 30\,\mathrm{ms}$, figure 3.8 of the simulated valve model can be made.

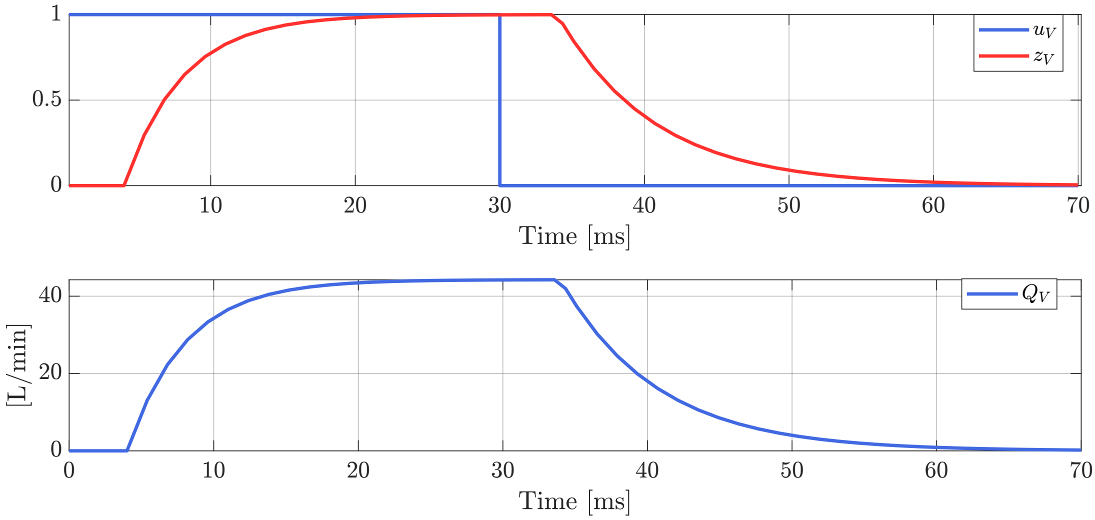


Figure 3.8. Valve test @ $p_1 = 100$ bar, $p_2 = 2$ bar and $T_d = 4$ ms

It is seen that there is a delay of 4 ms and a settling time of approximately 20 ms is achieved for the energizing state and similarly for de-energizing. It is clear that even though the valve is almost 100% open at 20 ms, it already permits significant flow much earlier (75% open @ 10 ms). Therefore it can not be enough to simply use the switching times of the data sheet to define the valve model, and it is expected that further tuning is necessary during model validation. The overall dynamics of the valve does however make sense and permits flow as expected and this concludes the valve modelling.

3.4 Chamber Pressures

The pressures in FBoT are modelled with continuity equations, one for each chamber.

$$\dot{p_C} = \frac{\beta_{eff}}{V_C}(Q_{in} - Q_{out} - \dot{V_C}) \tag{3.17}$$

The bulk modulus model is the same as the one presented for the supply system in section 3.2. The inflow and outflow correspond to the flow diagram. The volume of the chambers V_C are dependent on the piston position x_p and the dead volume V_{dC}:

$$V_C = \begin{cases} A_{set}x_p + V_{dC} & \text{Chambers 1 and 3} \\ A_{set}(L_s - x_p) + V_{dC} & \text{Chambers 2 and 4} \end{cases} \tag{3.18}$$

where A_{set} is the pressure area on the piston, and is equal for all chambers. The difference between the formulations of 1,3 and 2,4 lies in the definition of the origo of x_p. When the piston is at position zero, the chamber volume of chambers 2,4 are largest while chambers 1,3 are smallest, and vice versa when the piston is extended fully to length L_s.

From these volume definitions, the time dependent change in volume can be formulated for each chamber.

$$\dot{V_C} = \begin{cases} A_{set}\dot{x}_p & \text{Chambers 1 and 3} \\ -A_{set}\dot{x}_p & \text{Chambers 2 and 4} \end{cases} \tag{3.19}$$

The 4 continuity equations are:

$$
\begin{aligned}
\dot{p}_{C1} &= \frac{\beta_{eff}(p_{C1})}{A_{set}x_p + V_{dC}}(Q_{C1} - (Q_{le,C1T} + Q_{le,C1C3}) - A_{set}\dot{x}_p) \\
\dot{p}_{C2} &= \frac{\beta_{eff}(p_{C2})}{A_{set}(L_s - x_p) + V_{dC}}(Q_{C2} - (Q_{le,C2T} + Q_{le,C2C4}) - (-A_{set}\dot{x}_p)) \\
\dot{p}_{C3} &= \frac{\beta_{eff}(p_{C3})}{A_{set}x_p + V_{dC}}((Q_{C3} + Q_{le,C1C3}) - Q_{le,C3T} - A_{set}\dot{x}_p) \\
\dot{p}_{C4} &= \frac{\beta_{eff}(p_{C4})}{A_{set}(L_s - x_p) + V_{dC}}((Q_{C4} + Q_{le,C2C4}) - Q_{le,C4T} - (-A_{set}\dot{x}_p))
\end{aligned}
$$

From the equations it can be seen that chamber flows, leakage flows, volume change and fluid stiffness all affect how the pressures in the chambers are generated.

3.5 Piston Dynamics

This section treats how the piston will move due to forces subjected on it i.e. its equation of motion. These forces stem from the chamber pressures working on the piston areas along with friction.

3.5.1 Forces From Chamber Pressures

The force from the pressures is simply the sum of forces from the four different chamber pressures working on the piston

$$F_p = A_{set}p_{C1} + A_{set}p_{C3} - A_{set}p_{C2} - A_{set}p_{C4} \tag{3.20}$$

3.5.2 Friction Forces

The friction model takes basis from theory of viscous shear forces, which can be used in correlation to the free floating piston structure.

Shear stress of a Newtonian fluid is defined as

$$\tau_s = \mu\frac{d\dot{x}}{dy} \tag{3.21}$$

Forces from shear stresses are defined as

$$F_s = \tau_s A_s \tag{3.22}$$

Combining these expressions and by separation of variables and integration, the shear force is found as

$$F_s = A_s\mu\frac{\Delta\dot{x}}{h} \tag{3.23}$$

The viscosity of the fluid is kept constant, and the shear forces are therefore simply dependent on the piston position and speed. h is the gap between piston and chamber walls. To evaluate the friction forces, the geometry of the piston and housing must be considered. The piston is a piece-wise prismatic symmetric rod that can be seen to consist of 4 cylindrical parts. See figure 3.9 and 3.10 for illustrations with symbolic dimensions.

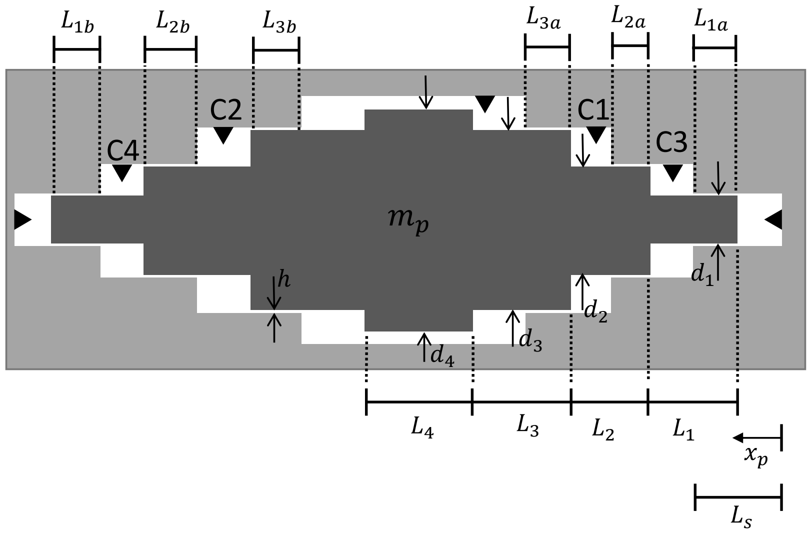


Figure 3.9. Piston and chamber geometry

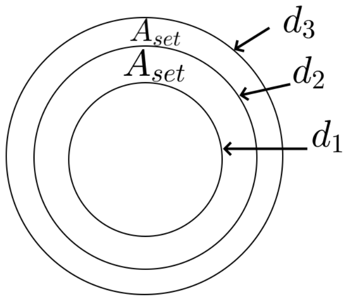


Figure 3.10. Relevant piston diameters

Since the area on each piston surface A_{set} is equal and predetermined, the piston diameters are based on this. The innermost diameter is set as $d_1 = 10\,\text{mm}$, and the other diameters can be calculated from the area of a circle as

$$A_{set} = \pi \left(\frac{d_2}{2}\right)^2 - \pi \left(\frac{d_1}{2}\right)^2 \Rightarrow d_2 = \sqrt{\frac{4}{\pi} A_{set} + d_1^2} \tag{3.24}$$

The lengths where the the piston overlaps with the chamber walls are dependent on the piston position as described in the left column below, and subsequently the overlaps between piston and chamber walls can be described as the right column:

$$\begin{aligned}
L_{1a} &= (L_1 - x_p) & A_{s,1a} &= d_1 \pi L_{1a} \\
L_{2a} &= (L_2 - x_p) & A_{s,2a} &= d_2 \pi L_{2a} \\
L_{3a} &= (L_3 - x_p) & A_{s,3a} &= d_3 \pi L_{3a} \\
L_{1b} &= (L_1 - (Ls - x_p)) & A_{s,1b} &= d_1 \pi L_{1b} \\
L_{2b} &= (L_2 - (Ls - x_p)) & A_{s,2b} &= d_2 \pi L_{2b} \\
L_{3b} &= (L_3 - (Ls - x_p)) & A_{s,3b} &= d_3 \pi L_{3b}
\end{aligned} \tag{3.25}$$

These areas are used in the shear force equation, where the Barus viscosity model is included. The pressure in the viscosity model is replaced with the average of the pressures on either side of the shear area, as an estimation of the pressure in the shear area.

$$F_{s,1a} = A_{s,1a}\mu_0 e^{\alpha_\mu(\frac{p_{C3}+p_T}{2})}\frac{\dot{x}_p}{h} \tag{3.26}$$

$$F_{s,2a} = A_{s,2a}\mu_0 e^{\alpha_\mu(\frac{p_{C1}+p_{C3}}{2})}\frac{\dot{x}_p}{h} \tag{3.27}$$

$$F_{s,3a} = A_{s,3a}\mu_0 e^{\alpha_\mu(\frac{p_{C1}+p_T}{2})}\frac{\dot{x}_p}{h} \tag{3.28}$$

$$F_{s,1b} = A_{s,1b}\mu_0 e^{\alpha_\mu(\frac{p_{C4}+p_T}{2})}\frac{\dot{x}_p}{h} \tag{3.29}$$

$$F_{s,ba} = A_{s,2b}\mu_0 e^{\alpha_\mu(\frac{p_{C2}+p_{C4}}{2})}\frac{\dot{x}_p}{h} \tag{3.30}$$

$$F_{s,3b} = A_{s,3b}\mu_0 e^{\alpha_\mu(\frac{p_{C2}+p_T}{2})}\frac{\dot{x}_p}{h} \tag{3.31}$$

The total shear force is the sum of the individual forces

$$F_s = F_{s,1a} + F_{s,2a} + F_{s,3a} + F_{s,1b} + F_{s,1b} + F_{s,1b} \tag{3.32}$$

3.5.3 Equation of Motion

The forces act on the piston, and the acceleration is found via Newton's II law

$$\ddot{x}_p = \frac{F_p - F_s}{m_p} \tag{3.33}$$

Where m_p is the mass of the piston. The acceleration is doubly integrated to find position and velocity during simulation. This simple equation of motion emphasizes the simplicity of the FBoT mechanical system.

3.5.4 Piston Simulation

A valve model is combined with two chamber pressure equations and the piston equation of motion to evaluate whether the model behaves as expected. Chamber 3 and 4 are connected to tank, chamber 2 is locked, chamber 1 is connected to the supply through an open valve, but the supply system is simplified to a constant pressure such that the piston simulation is isolated. Leakage is not included. The results are seen in figure 3.11

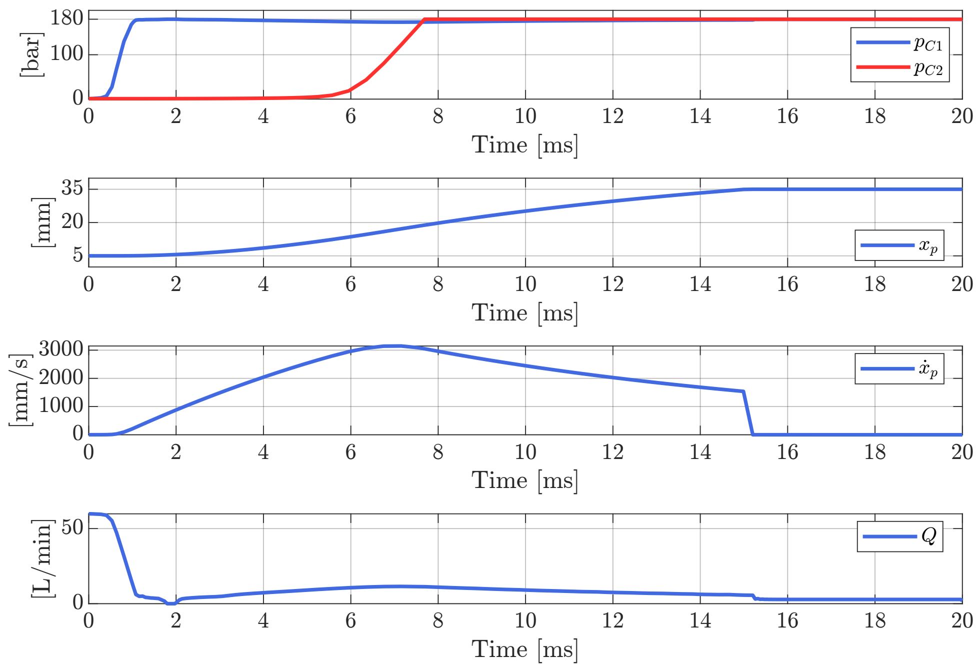


Figure 3.11. Piston simulation

Both chamber 1 and 2 pressures start at tank level. Due to the pressure difference between chamber 1 and supply, a large flow Q enters chamber 1. This flow makes the chamber pressure rise, and since chamber pressure 1 is larger than chamber 2, the piston begins to accelerate. The acceleration does however happen relatively slowly, so chamber 1 pressure continues to rise rapidly until it reaches the same level as the supply pressure, at which point the flow diminishes. From time 2 ms until 4.5 ms, the pressure difference is almost constant, and the velocity of the piston increases seemingly linearly i.e. the acceleration is constant, which is as expected. Friction does however slow it down. Chamber 1 pressure only drops ever so slightly due to expansion of chamber 1 volume, since the supply flow is ever present. At time 6 ms chamber 2 pressure starts to rise rapidly solely due to compression of chamber 2, and during the time 6-8 ms the piston decelerates since the pressure difference between the two chambers diminishes. At time 8 ms the two chamber pressures are practically equal (chamber 2 is overflowing through the check valve into the supply at supply pressure, and chamber 1 pressure is slightly lower since the chamber volume is still expanding). When the pressure difference is zero, the net force on the piston should ideally also be zero, and the piston speed should be constant, however, the speed is dropping and this is due to the friction forces. The friction forces are slowing the piston down gradually, but when the piston reaches the end-stop at 35 mm the speed naturally goes to zero. The piston dynamics seem to perform as could be expected, but will possibly require tuning during the model validation, especially in regards to the viscosity parameter which directly influences the friction.

3.6 Leakage Flows

Due to the nature of the free floating piston, there are inherent leakage flows. These leakage flows will be modeled as laminar flows. Laminar flow in a small gap can be described as

$$Q = \frac{h^3 w}{12 \mu L} \Delta p \tag{3.34}$$

Where h is the gap, w is the width of the gap, L is the length of the gap and Δp is the pressure difference on either side of the gap as illustrated on figure 3.12 *a)* and *b)*. The piston and chamber are cylindrical, but when the diameter of the piston is much larger than the size of the gap, the circumference of the piston including half of the gap can be used as the width. That is, according to figure 3.12 *b)*:

$$w = \pi(d + D)/2 = \pi(d + (d + 2h))/2 = \pi(2d + 2h)/2 = \pi(d + h) \tag{3.35}$$

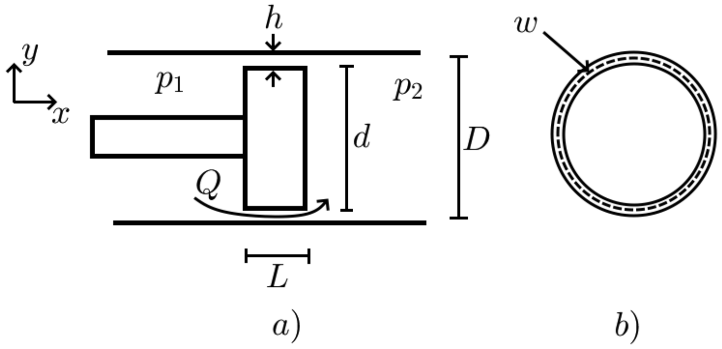


Figure 3.12. Leakage Flow Diagram

Reusing the length definitions of the piston and chambers from previous section, the 6 leakage flows are modelled, including the Barus viscosity equation, as:

$$Q_{le,C1T} = \frac{h^3\pi(d_3 + h)}{12\mu_0 e^{\alpha_\mu(\frac{p_{C1}+p_T}{2})} L_{3a}}(p_{C1} - p_T)$$
$$Q_{le,C1C3} = \frac{h^3\pi(d_2 + h)}{12\mu_0 e^{\alpha_\mu(\frac{p_{C1}+p_{C3}}{2})} L_{2a}}(p_{C1} - p_{C3})$$
$$Q_{le,C3T} = \frac{h^3\pi(d_1 + h)}{12\mu_0 e^{\alpha_\mu(\frac{p_{C3}+p_T}{2})} L_{1a}}(p_{C3} - p_T)$$
$$Q_{le,C2T} = \frac{h^3\pi(d_3 + h)}{12\mu_0 e^{\alpha_\mu(\frac{p_{C2}+p_T}{2})} L_{3b}}(p_{C2} - p_T)$$
$$Q_{le,C2C4} = \frac{h^3\pi(d_2 + h)}{12\mu_0 e^{\alpha_\mu(\frac{p_{C2}+p_{C4}}{2})} L_{2b}}(p_{C2} - p_{C4})$$
$$Q_{le,C4T} = \frac{h^3\pi(d_1 + h)}{12\mu_0 e^{\alpha_\mu(\frac{p_{C4}+p_T}{2})} L_{1b}}(p_{C4} - p_T)$$

The leakage model is tested. figure 3.13 shows all leakage flows for a varying piston position.

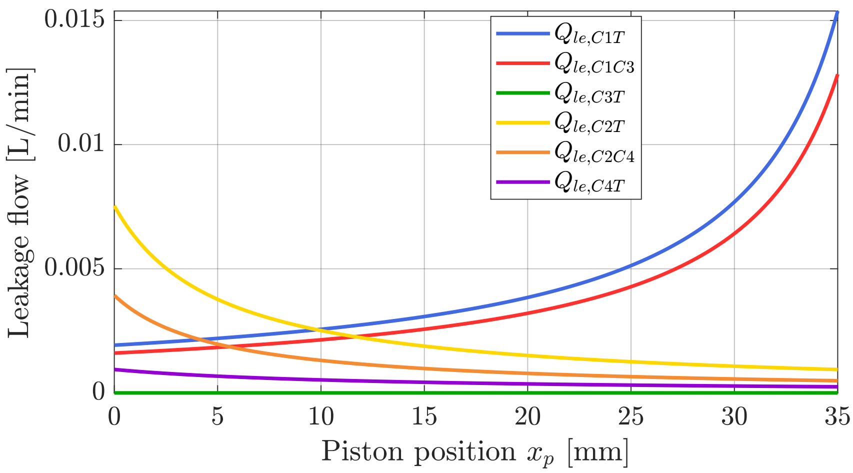


Figure 3.13. Chamber leakages at different piston positions. @ pC1 = 100 bar, pC2 = 50 bar, pC3 = 2 bar, pC4 = 20 bar

It is seen that the leakages grow/decay seemingly exponentially for a varying piston position. Leakages at the right side of the piston grow when the piston extends, and the left side leakages decay, which is as expected since the laminar flow distance is respectively shortened and extended.

3.7 Load Model

The load side of the FBoT leads to a load cylinder chamber in the model. In reality, the cylinder drives the boom of the wheel loader which will endure a large variety of loads, all of which could be modelled to some degree by considering the geometry of the wheel loader and the kinematics. But different load cases will only result in a different load force on the cylinder and a different inertia, and therefore the load model will be simple and only considered as a cylinder with a load, where it in the simulation will be possible to change the load force and inertia arbitrarily to evaluate the controller's robustness.

The pressure in cylinder chamber A is modelled with the continuity equation as:

$$\dot{p}_A = \frac{\beta_{eff}}{V_L}(Q_{in} - \dot{V}_L) \Rightarrow \dot{p}_A = \frac{\beta_{eff}}{A_A x_c + V_{dCA}}(Q_L - A_A \dot{x}_c) \tag{3.36}$$

Where the bulk modulus model is similar to the one presented in the supply model section 3.2. The initial model parameters such as cylinder piston areas A_A and A_B and dead volume V_{dCa} are somewhat uncertain. It is assumed that the B chamber is connected to the tank i.e. the pressure is constant at tank level $p_B = p_T$, hence a continuity equation is not necessary.

The movement of the cylinder piston is modelled using Newton's II law as:

$$\ddot{x}_c = \frac{p_A A_A - p_B A_B - B_{cyl}\dot{x}_c - F_{load}}{m_c} \tag{3.37}$$

Where a viscous damping term $B_{cyl}\dot{x}_c$ is included.

3.8 Model Validation

A mathematical model of the system has been developed and implemented in Simulink, but various parameters are uncertain, so the model must be validated. There have been produced some test data at the lab setup which will be analysed and compared with simulations of the model.

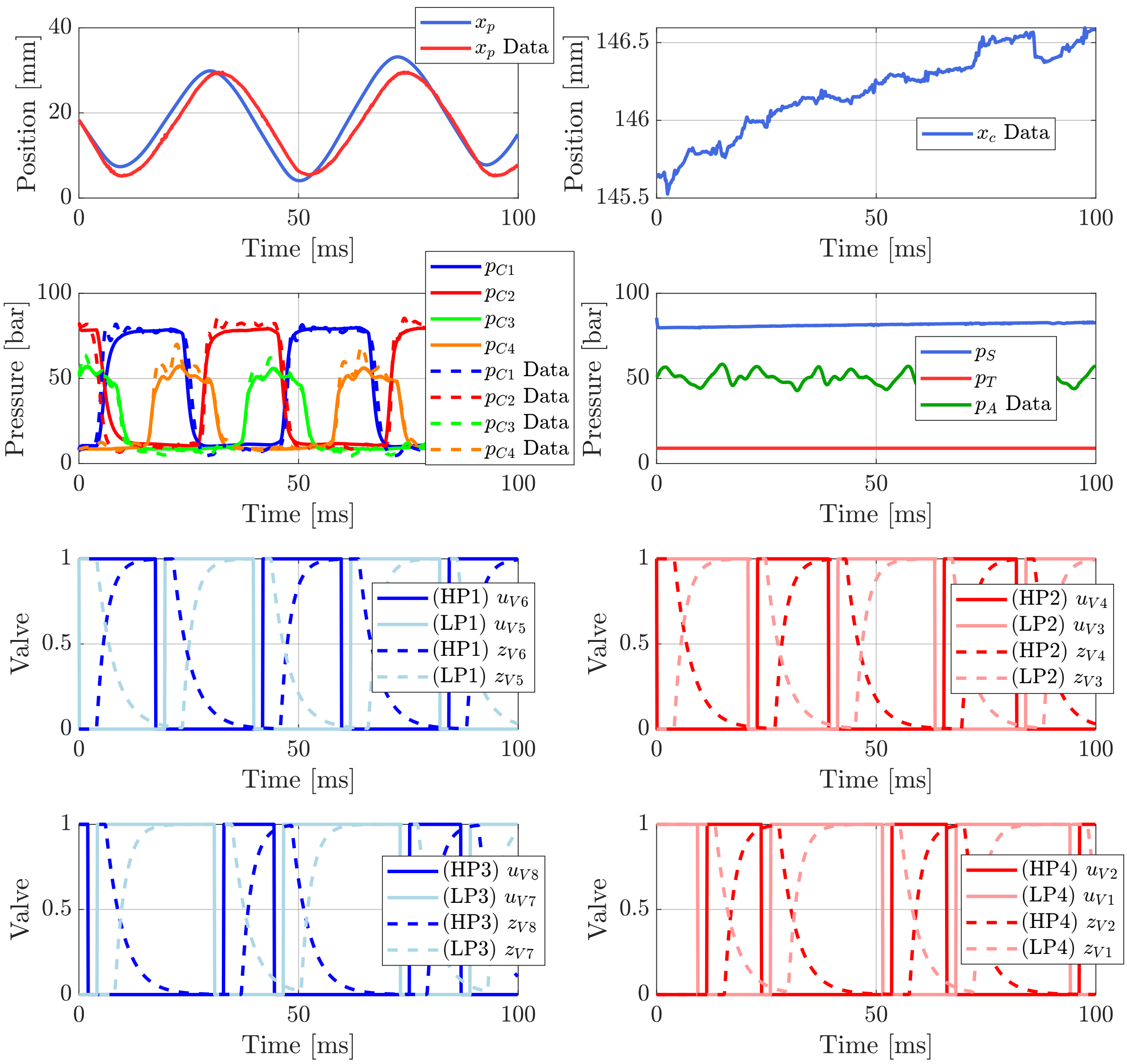


Figure 3.14. Data from FBoT test 1, supply to load during 100 milli-seconds

The input valve signals from the data set are used as inputs to the Simulink model, and initial conditions are matched with the data set. The load model is replaced with load data to make the conditions of the model as representative as possible. That means that the load pressure in the model will be the time series pressure in the load chamber from the data set. The dynamic behaviour of the load cylinder will thus not be captured, but since the load model is not corresponding the lab setup, then it is expected that the valve control signals will not match very well. The initial volume of the accumulator fluid is adjusted such that the accumulator pressure matches the initial supply pressure. The reference supply pressure follows the supply pressure from the data. The tank pressure is adjusted to a higher value of 9 bar. The valve dynamics are adjusted such that the time delay is $T_d = 4\,\text{ms}$, the desired settling times for respectively energizing and de-energizing are $(6 + 20)/2[ms]$ and $(10 + 30)/2[ms]$. The other dynamic parameters are calculated from equation 3.16. Other parameters are set as $\mu_0 = 0.4002$, $\alpha_\mu = 2.00 \times 10^{-10}$, $\alpha_{air} = 2\%$.

The simulation data is superimposed with the data as seen in figure 3.14. The four bottom subplots show the valve control signals as the solid line and actual position as dashed line. There is no data of the actual valve positions, so that is only model, but the control signal is equal. In the legend of the valve figures, there are included a parenthesis e.g. (HP1) which means High Pressure chamber 1, and (LP1) means Low Pressure chamber 1, for ease of reading the graphs. The control signals have been generated from the control system that was implemented on the lab setup during the experiment. These signals results in the correct timing of pressurizing and de-pressuring the FBoT chambers such that the piston oscillates while feeding oil into the load chambers. The sequence of control signals stem from a preliminary control system which will not be further elaborated. The top left figure shows the piston position, and it is seen that it indeed oscillates. The lab data behaves more consistent than the model. It seems that the model at first reacts a bit faster than the lab data and the model leads the lab data throughout the simulation. The viscosity has been increased, such that the speeds matches. It should be kept in mind that the control signals are generated with piston feedback, so a small discrepancy of the piston model will very quickly escalate to large errors, as can be seen. Therefore it does not make sense to tune the model to "perfection" and it will practically be impossible. The most important thing is that the model is so accurate that it is rendered probable that it can be used to develop controllers. top right is load cylinder position, although that is not of particular interest as explained before, since the load cylinder model has been replaced solely by data. It shows a little noisy data, and that the movement is not very smooth, although it should be noticed that the entire movement happens within 100 ms in the figure. Sub figure (2,1) i.e. second row left shows the pressures in the chambers surrounding the piston. The model results are shown as solid lines and the lab data is shown as dashed lines. Regarding the pressure timings, the model behaves very good i.e. the trajectories are seemingly equal. There are small differences e.g. the model does not capture pressure oscillations and the pressures in chambers 3-4 are larger in the lab data, which might be due to some unmodelled hose dynamics where the pressures needs to be larger to initiate flow. That the pressure in chamber 4 is a little too small might very well be the reason that the piston travels too fast. The main aspects showing that the model can be good enough for controller design is that the chamber pressures behave faithfully according to the valve control signals, and that the piston accelerates and moves realistically due to chamber pressures. Therefore the model is deemed as validated and useable for controller design.

Chapter 4
Control Focus

In this chapter the overall control objective is explored. First, the working principles and desired properties are considered which results in a problem formulation and control requirements.

4.1 Working Principle

The fundamental purpose of the FBoT is to transform fluid power efficiently. Through energy conservation, without considering losses, this translates to

$$p_I V_I = p_O V_O \tag{4.1}$$

where p_I and p_O are the input and output pressure respectively, and V_I and V_O are the input and output displaced volumes respectively. This relation shows that for any pressure combination, there is a solution for how much volume should be displaced to conserve the energy:

$$\frac{p_I}{p_O} = \frac{V_O}{V_I} \tag{4.2}$$

The intended working principle of the FBoT is to achieve this transformation by converting the energy of an input to kinetic energy in the piston which is converted to output energy. This principle is illustrated in figure 4.1. Note that the figure does not represent the actual FBoT and is only for illustration. In this case, the input and output pressures are equal. From time zero to 0.5, the input pressure is p_1 and the piston accelerates due to work from the input pressure displacing a volume. At time 0.5, the input pressure stops, and the output pressure rises due to compression by the piston. During this compression, the kinetic energy is converted into displaced volume at the output pressure. At time 1, the piston is at standstill, and p_2 rises due to compression of the chamber which simultaneously acts as a braking pressure. p_2 becomes the input pressure and the piston starts accelerating. At time 1.5 p_2 falls, and p_3 (new output pressure) rises. At the very end, p_1 rises again and will act as the input pressure for the next cycle. The graph shows two half-cycles, and it can be seen that the first half-cycle is similar to the other, except that the roles of the pressures have switched.

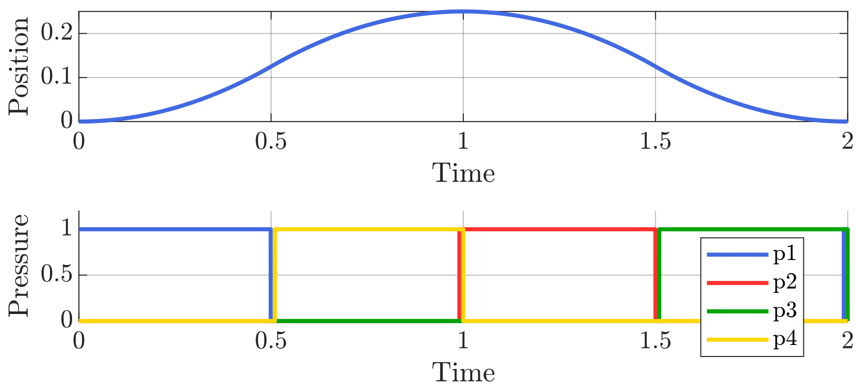


Figure 4.1. FBoT basic working principle

The intended working principles can be stated as a problem formulation, which will focus the remainder of the report on control design:

"How can a control system be developed such that the intended working principles of the FBoT are achieved?"

In accordance with the intended working principles, the control requirements for the FBoT can be formulated, for the purpose of designing a control framework:

- The piston must move reciprocally within positional constraints
- Input flow must be converted to an output flow during piston movement
- To maintain high efficiency, on/off valves can only open when the pressure difference across the valve is sufficiently low
- To maintain high efficiency, there must never be a direct connection between a high pressure manifold and tank
- The FBoT must function in both pump mode and regenerate mode which correspond to transforming fluid power from supply to load, and from load to supply respectively
- It must be possible to modulate average output flow i.e. a fixed output flow for some manifold pressure case is not acceptable. It must be possible to vary the output flow between zero and some maximum potential, such that the FBoT can be used in applications where precise boom movement is required as an example.

4.2 Control Strategy Concept

To achieve the intended working principle of the FBoT is not as simple as implementing a generic PID controller or the like, since the inputs to the system are discrete states of valve being on or off and the input sequence is dependent on the operating mode and furthermore, the controller needs to know when to activate inputs i.e. have some kind of future knowledge. These considerations inspires the use of Model Predictive Control (MPC) since this approach uses a system model to anticipate future events and optimize the inputs such that constraints are not violated and that the desired trajectory is followed. MPC generally requires On-line control and powerful computational hardware.

Within this overall branch of control design, it is possible to generate control strategies. An initial idea could be to control the reference frequency of the piston trajectory as a way to modulate the output flow. This approach does however not consider that there may only be a single most efficient frequency for a certain input/output pressure combination, which can make this approach inefficient. A related control idea is to not control the piston frequency, but to instead control the piston amplitude. The idea is that this will alter the average load flow, but the same cons apply here as for the reference frequency approach. To modulate the average flow, it is most likely more efficient to use full strokes and apply a type of pulse density modulation (PDM). Methods have been developed by Johansen et al. (2015) in regards to PDM for digital hydraulics, and similar methods are suitable for the FBoT but further PDM development is beyond the scope of this report, and will only be treated superficially.

By using PDM a good control approach is to make each half-cycle as efficient as possible while maximizing the load flow. The basic idea is simple: Input pressure accelerates the piston, and the output pressure absorbs the kinetic energy of the piston again, and finally a pressure will make sure the piston stops completely while developing pressure for the next half-cycle, - which is exactly like the previously described working principle, but the control problem is to find the most optimal switching times of the functional pressures, which will require optimization to realize with a nonlinear system with losses.

Chapter 5
MPC Design

The overall control strategy has been chosen to be within a MPC framework. The purpose of this chapter is to present and explain how it works and elements of the design process.

5.1 Working Principle

The MPC design differs slightly from classical formulations, where manipulated variables can be independent from each other at each time instant (García et al., 1989). This is not applicable for control of the FBoT since the inputs to the system are valve opening signals, and these are not allowed to turn on and off freely and cannot be open halfway i.e. proportionally. Instead, it is more useful to manipulate the switching times of the valves, which however adds another dimension to the control problem. The system inputs are no longer controlled directly, but will be a function of switching times. This means that off-the-shelf solutions such as Matlab's MPC toolbox cannot be used. The rest of this chapter aims to formulate a unique MPC formulation that embeds switching times as design variables.

A model predictive controller does not state an explicit control law, but produces control signals based on optimization for each step in time. A prediction is made for a finite time horizon, where the switching times are optimized. The switching times are converted to a series of input signals spanning the prediction horizon, but only the very first inputs are used on the plant. The control structure of the MPC is seen in figure 5.1. A user supplies some inputs which are processed in the controller. Mode refers to whether the FBoT should pump or regenerate, and average flow modulation (AFM), which will be elaborated in the following subsection. These user inputs are fed into the controller along with estimated plant states $\hat{\boldsymbol{q}}$. The controller uses optimization in combination with the prediction model to produce optimal inputs to the plant $\boldsymbol{u}$. The plant reacts from inputs and its own states to produce new states $\boldsymbol{q}$. Some of the states can be observed with an output matrix to get outputs $\boldsymbol{y}$, from which the systems states are estimated with an observer.

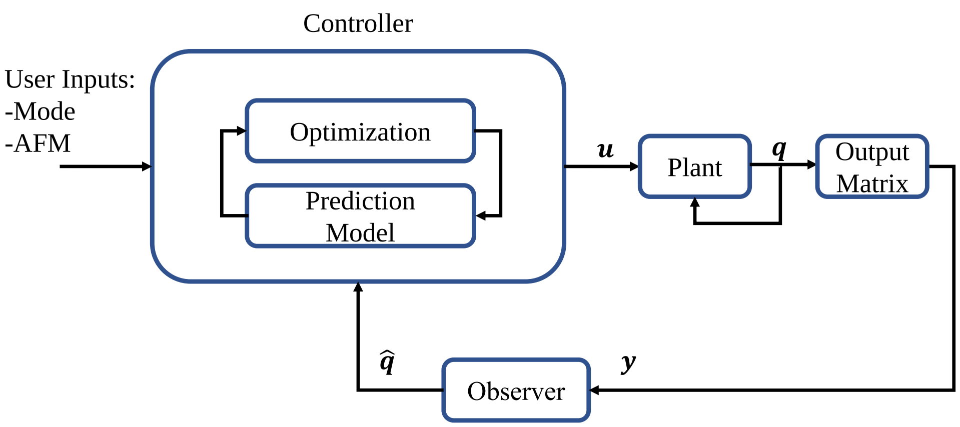


Figure 5.1. Control Diagram

5.1.1 User Inputs

The FBoT mode has been elaborated earlier and refers to whether it should function in pump mode, where power is transformed from the supply to the load, or function in regeneration mode where power is transformed from the load to the supply. The other input is the average flow modulation which refers to the desired average output flow as a percentage. 100% refers to maximum flow potential where every piston stroke transforms power and outputs flow and there are no pauses between strokes. For every other percentages, there will be a combination of full power strokes and pauses such that the average flow is decreased. This aspect will not be explored further through the report.

5.1.2 Prediction Horizon

The primary feature of MPC centers on the concept of simulating a model for a prediction horizon and using the knowledge to produce optimal inputs. The control problem considers the model simulated for a certain time period, which is called the prediction horizon in MPC terminology or receding horizon. Receding horizon refers to the working principle that the prediction horizon is moved for every control time step. The prediction horizon spans a local time frame from time 0 to T_p in time steps k_p with a sample time T_{ps}. It is important to note the difference between local time and global time when working with MPC. The local time refers to a single prediction horizon, which locally spans from 0 to T_p. The global time is normal, real time where actual control is implemented - either on a physical system or virtually. The global time is tracked in time steps k starting from time 0 and spans forever or until control is turned off (T_{end}). The local time's starting point (time zero) is always at time step k, and the prediction horizon can thus globally be seen as spanning from k to $k + k_p$.

Figure 5.2 shows the basic principle of the MPC scheme for a single prediction horizon. During this prediction horizon, the model is simulated with the given switching times which will result in a time series of piston trajectory shown in the top part of the graph, while the inputs to the system are shown in the bottom part. Here it can be seen that the system inputs are valve signals being either 1 or 0, and when the signals are active are determined by switching times. See for example $t_{I,HP,on}$ which dictates when the High Pressure (HP) valve to the input chamber is opened, and $t_{I,HP,off}$ dictates when the valve must close. Similar switching times are dictated for output and braking pressures which will be elaborated in the subsequent section.

5.1.3 Switching Times

To achieve the pressures and piston positions as sketched in figure fig. 4.1 on page 22, the inputs i.e. the on/off valves must be turned on and off in a specific order and at specific times. These switches are defined as switching times. The fewest amount of switches to define an input for a given horizon, except a trivial solution, must be two: switch on and switch off, or oppositely switch off and switch on. If there is only one switch, then the switching time risks changing function between predictions (e.g. from turning on input to turning off input) which is an unnecessary complication. In future works it may be possible to reduce the control problem to fewer switching times, but in this control proposal, there will be two switching times per input signal, which gives a total of 16 switching times for a single prediction horizon. The relationship between each switching time and input on/off is given in table 5.1. From the table it is seen that switching time t_1 controls when input u_1 turns on, and what function it has, namely to connect chamber 1 to its high pressure manifold. It is noted that switching times for valves u_5 to u_8 are switched around. These valves are open by default and the switching times dictate when the

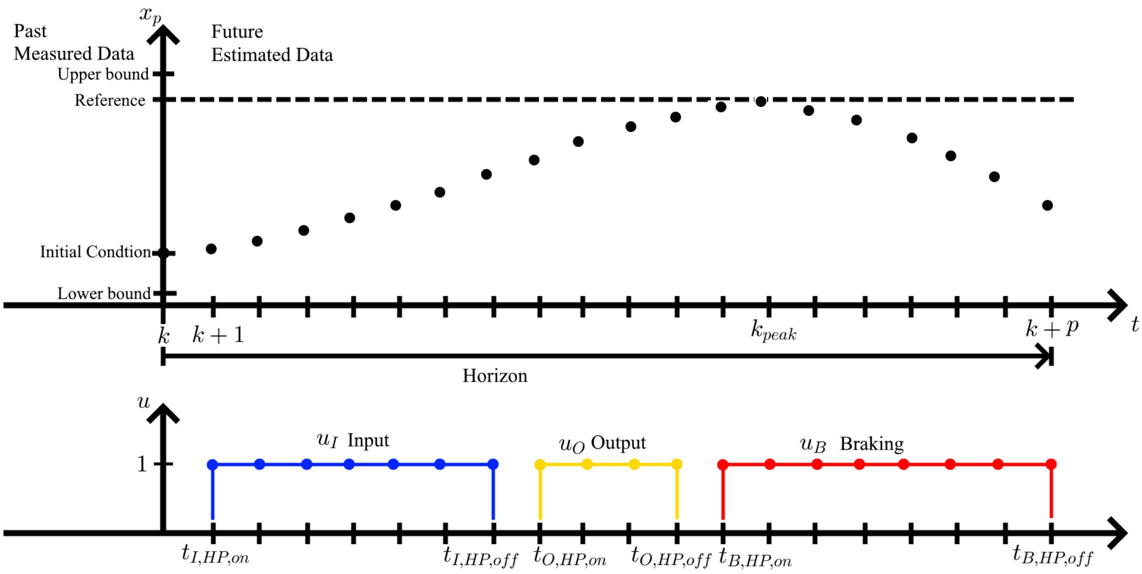


Figure 5.2. Prediction horizon

valves must close off and open again. These relationships can be seen graphically in figure 5.3. Each quadrant of the figure represents one of the four FBoT chambers.

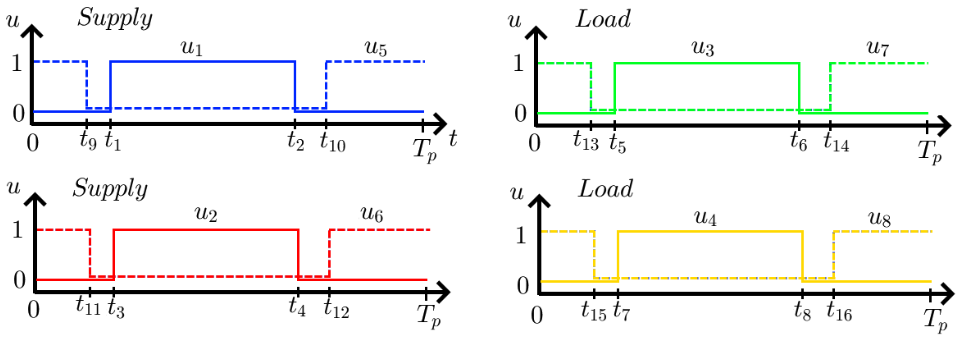


Figure 5.3. Switching times and inputs relationship

5.1.4 Chamber Roles

It is possible to design the MPC such that all 16 switching times are design variables, but the problem can effectively be simplified. As indicated in figure fig. 4.1 on page 22, the four FBoT chambers changes roles depending on the specific working case. There are two operating modes: 1. pump mode which transforms flow from supply to load manifold, and 2. regeneration mode which transforms flow from load to supply manifold. The FBoT does not care about the difference between supply and load manifold, since both sides are high pressure manifolds of which the fluid can flow in or out of. The distinction is only what manifold works as input and which works as output. For pump mode, the two supply chambers 1 and 2 function as input, and load chambers 3 and 4 function as outputs. For regenerate mode the opposite is true. But for either mode, there are two choices between input chambers (and output chambers), e.g. in pumping mode, both chamber 1 and 2 work as input. The choice between input chamber can be made by reflecting on which direction the piston should move at a given point i.e. what is its *target*. For positive directions in pumping mode, chamber 1 must act as the input, and reversely chamber 2 for negative directions. Refer to the hydraulic diagram in figure fig. 2.3 on page 6 for definition of positive piston direction. This gives a total of four operating cases, where the input

Switching Time	Input	State	Chamber Pressure
t_1	u_1	On	HP Chamber 1
t_2	u_1	Off	
t_3	u_2	On	HP Chamber 2
t_4	u_2	Off	
t_5	u_3	On	HP Chamber 3
t_6	u_3	Off	
t_7	u_4	On	HP Chamber 4
t_8	u_4	Off	
t_9	u_5	Off	
t_{10}	u_5	On	LP Chamber 1
t_{11}	u_6	Off	
t_{12}	u_6	On	LP Chamber 2
t_{13}	u_7	Off	
t_{14}	u_7	On	LP Chamber 3
t_{15}	u_8	Off	
t_{16}	u_8	On	LP Chamber 4

Table 5.1. Switching Times corresponding to inputs turning on or off

and output chambers can be defined. As an extra role, a braking pressure is defined as the input chamber for the following half-cycle. This chamber must be controlled such that high pressure can be built up before the piston reaches the end of its trajectory to ensure smooth and effective operation. It is called a braking pressure since its pressure provides negative force compared to the input i.e. the pressure brakes the piston while it does not act as input.

Having given each chamber a role for the four operating cases, it is possible to do the same for the switching times that control each chamber. All roles for every case is collected in table 5.2. The corresponding manifold high pressure (HP) values for each role is added along with switching times design variables identifications $\boldsymbol{t}_{des,id}$. It can be noted that the two entries in the design variable vector are the same as the input HP turn off time and braking HP turn on time respectively. This will be explained more in depth in section 5.3.1. Positive piston directions refers to ***Target*** being **High** and vice versa for **Low**.

The table contains extensive information about roles. The table is divided into 4 portions, one for each case of mode and target combination. Within each of these portions, the pressure roles are assigned. As an example the top left quadrant which is for pumping mode and high target is considered. The input pressure role p_I is assigned to chamber pressure p_1, and the corresponding manifold high pressure $p_{I,HP}$ is assigned to the supply manifold pressure p_S. The same is done for output pressure and braking pressure. In the same portion, the switching time roles are assigned. The switching times are collected in vectors containing the switching time for turning on and turning off a valve represented as $\boldsymbol{t}_{I,HP,[on,off]} = [t_1 \; t_2]$ as an example. This can also be written as $t_{I,HP,on} = t_1$ and $t_{I,HP,off} = t_2$ which means that valve switching time t_1, which is when the HP valve of chamber 1 is turned, is assigned to the role of $t_{I,HP,on}$ which is the switching time of when the input HP valve turns on. Figure fig. 5.3 on the facing page shows which valve switching times correspond to which functions for each chamber.

	Mode					
	Pump			**Regenerate**		
Target	*Pressure Role*	*Switching Time Role*		*Pressure Role*	*Switching Time Role*	
High	$p_I = p_1$	$\boldsymbol{t}_{I,HP,[on,off]}$	$[t_1 \quad t_2]$	$p_I = p_3$	$\boldsymbol{t}_{I,HP,[on,off]}$	$[t_5 \quad t_6]$
	$p_{I,HP} = p_S$	$\boldsymbol{t}_{I,LP,[off,on]}$	$[t_9 \quad t_{10}]$	$p_{I,HP} = p_L$	$\boldsymbol{t}_{I,LP,[off,on]}$	$[t_{13} \quad t_{14}]$
	$p_O = p_4$	$\boldsymbol{t}_{O,LP,[off,on]}$	$[t_{15} \quad t_{16}]$	$p_O = p_2$	$\boldsymbol{t}_{O,LP,[off,on]}$	$[t_{11} \quad t_{12}]$
	$p_{O,HP} = p_L$	$\boldsymbol{t}_{B,HP,[on,off]}$	$[t_3 \quad t_4]$	$p_{O,HP} = p_S$	$\boldsymbol{t}_{B,HP,[on,off]}$	$[t_7 \quad t_8]$
	$p_B = p_2$	$\boldsymbol{t}_{B,LP,[off,on]}$	$[t_{11} \quad t_{12}]$	$p_B = p_4$	$\boldsymbol{t}_{B,LP,[off,on]}$	$[t_{15} \quad t_{16}]$
	$p_{B,HP} = p_S$	$\boldsymbol{t}_{des,id}$	$[t_2 \quad t_3]$	$p_{B,HP} = p_L$	$\boldsymbol{t}_{des,id}$	$[t_6 \quad t_7]$
Low	$p_I = p_2$	$\boldsymbol{t}_{I,HP,[on,off]}$	$[t_3 \quad t_4]$	$p_I = p_4$	$\boldsymbol{t}_{I,HP,[on,off]}$	$[t_7 \quad t_8]$
	$p_{I,HP} = p_S$	$\boldsymbol{t}_{I,LP,[off,on]}$	$[t_{11} \quad t_{12}]$	$p_{I,HP} = p_L$	$\boldsymbol{t}_{I,LP,[off,on]}$	$[t_{15} \quad t_{16}]$
	$p_O = p_3$	$\boldsymbol{t}_{O,LP,[off,on]}$	$[t_{13} \quad t_{14}]$	$p_O = p_1$	$\boldsymbol{t}_{O,LP,[off,on]}$	$[t_9 \quad t_{10}]$
	$p_{O,HP} = p_L$	$\boldsymbol{t}_{B,HP,[on,off]}$	$[t_1 \quad t_2]$	$p_{O,HP} = p_S$	$\boldsymbol{t}_{B,HP,[on,off]}$	$[t_5 \quad t_6]$
	$p_B = p_1$	$\boldsymbol{t}_{B,LP,[off,on]}$	$[t_9 \quad t_{10}]$	$p_B = p_3$	$\boldsymbol{t}_{B,LP,[off,on]}$	$[t_{13} \quad t_{14}]$
	$p_{B,HP} = p_S$	$\boldsymbol{t}_{des,id}$	$[t_4 \quad t_1]$	$p_{B,HP} = p_L$	$\boldsymbol{t}_{des,id}$	$[t_8 \quad t_5]$

Table 5.2. Pressure Role dependent on mode and target

5.1.5 Global Time MPC Implementation Structure

This subsection aims to explain how the MPC works in combination with the plant figuratively.

At time instant k the plant is measured and the states serve as initial conditions for the prediction model. The prediction model is then simulated for its horizon producing a time series of system states. An optimization scheme changes the design variables which are the switching times such that a cost function is minimized while keeping the trajectory within the constraints. When an optimal solution has been found, the input from the prediction horizon at local time 0 (global time k) is fed into the plant and global time progresses a single time step. Then the whole optimization starts over with updated initial conditions.

Figure fig. 5.4 on the next page shows predictions in comparison to plant implementation. The top graph shows the desired trajectory of the piston along with chamber pressures and input signals. The top graph consists of 6 sub-graphs which shows piston trajectory, chamber pressures and valve signals for the four chambers. Each valve signal graph contains the signal for high pressure valve signal and the low pressure signal. The three shorter graphs below show prediction horizons initiated at different points in time. Each of these consist of three sub-graphs which shows predicted piston trajectory, predicted chamber pressures as pressure roles, and valve signals as roles. The valve signal sub-graph contains 6 valve signals: Input, Output and Braking HP and LP signals. Note that predictions are made at every single global time step, and this figure only serves to show the working principles, assuming a perfect optimization has been implemented. In a single

The first prediction (a.) starts from initial values, its reference position is *high* and the input is chamber 1, output is chamber 4 and braking pressure comes from chamber 2. The prediction finds optimized design variables, and the first time step of the prediction horizon is implemented. After some time, prediction (b.) is shown. Here, the input pressure has been through a cycle of being off, on and off again, and it cannot be activated again before x_{ref} has been reset (See more in section 5.3.1). The output and braking pressure is meanwhile still working. There might be a difference between the predicted peak time and the actual peak time, which is a natural part of an MPC implementation - The prediction model used for optimization will never be perfect, which is why the input trajectory constantly needs to be recalculated and updated to accommodate for model errors, disturbances and other factors that will introduce infidelity.

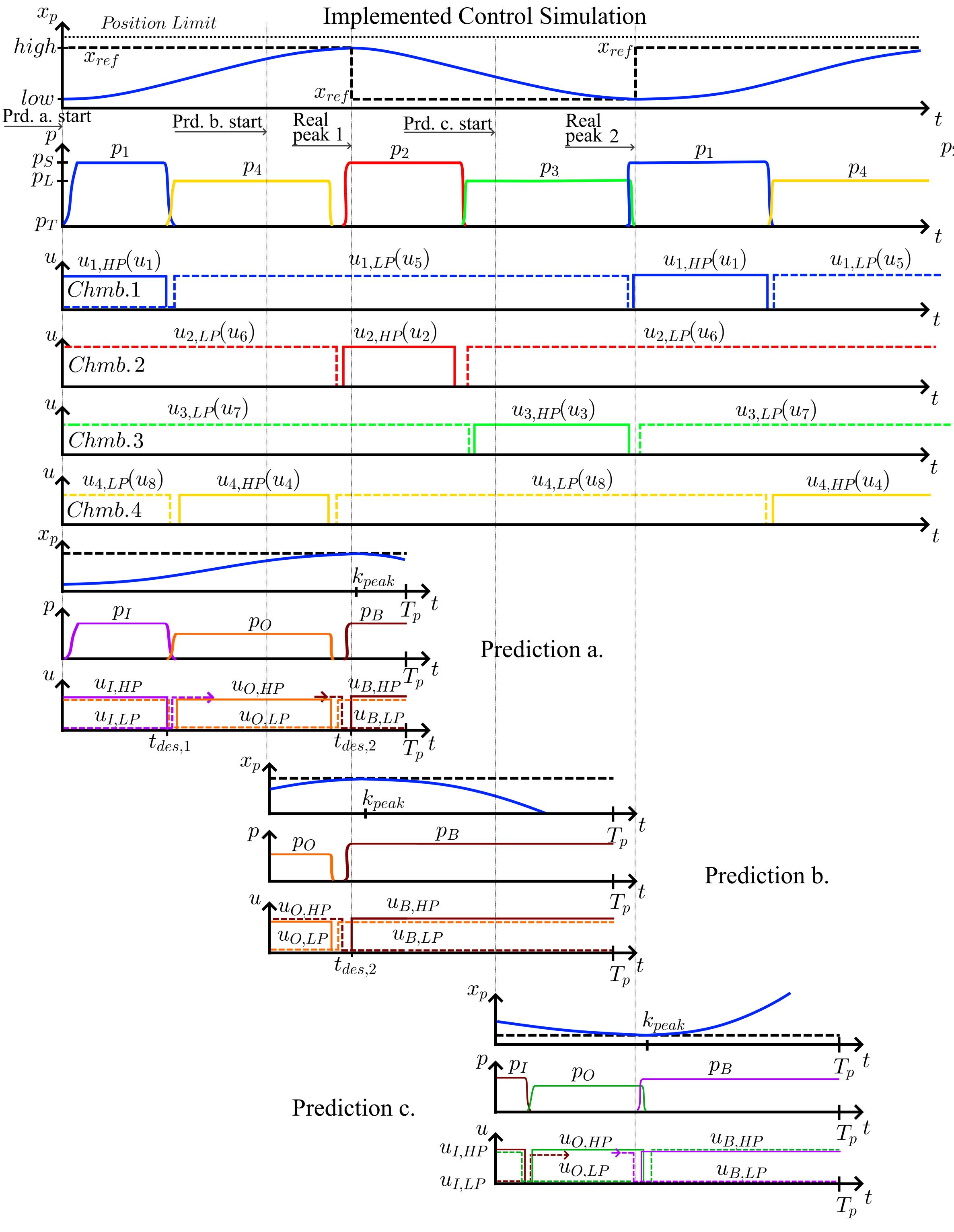


Figure 5.4. MPC Desired Functioning

It can be seen for prediction (b.) that the braking pressure never turns off. This is because its HP turn off point will always be the end of the prediction horizon. This has the effect that the braking pressure stays on, and the piston keeps accelerating beyond its constraints. However, this is acceptable, since constraints are only evaluated until the local piston position peak. What happens afterwards is irrelevant since the corresponding input signals never will be implemented. When the real peak is reached, the roles of the pressure chambers are switched, and the braking pressure will become the input pressure which can be turned off. This is seen in prediction (c.) where x_{ref} has become *low*. The new output is chamber 3 and the braking pressure is in chamber 1. This can also be seen by the design variables that have reappeared in a configuration that is similar to that in prediction (a.). This shows the cyclic nature of the control strategy - chambers shifting roles dependent on target reference and mode (pump/regenerate). Regenerate mode is not shown in the graph but the only difference is again the roles of the chambers, which can be read off table 5.2. This approach also allows for pressure transformations from high to low, and low to high which gives this control proposal versatility.

5.2 Prediction Model

This section will elaborate on the details of the prediction model. The model presented in chapter 3 is far too complicated to be used for optimization in connection with MPC, since 1. there needs to be made a large number of predictions and 2. each prediction must be very fast. The full Simulink model takes a few seconds to evaluate on an ordinary laptop PC, which is too slow even considering that the control hardware on the lab setup is much faster. The prediction model should be simulated within less than a millisecond for the MPC approach to be feasible, and this section aims to develop a reduced surrogate model.

5.2.1 Model Inputs

There are 8 on/off valves which are inputs to the system. Vector $\boldsymbol{u}$ contains the the 8 input values.

$$\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \end{bmatrix}^T$$

Each entry can be either 1 or 0

$$u_i \in [1, 0]$$

Each valve corresponds to a function for each chamber as described in table 5.3. The table essentially restates what can be read of the hydraulic diagram fig. 2.3 on page 6.

u_1	Chamber 1 HP - Connect Chamber 1 to Supply
u_2	Chamber 2 HP - Connect Chamber 2 to Supply
u_3	Chamber 3 HP - Connect Chamber 3 to Load
u_4	Chamber 4 HP - Connect Chamber 4 to Load
u_5	Chamber 1 LP - Connect Chamber 1 to Tank
u_6	Chamber 2 LP - Connect Chamber 2 to Tank
u_7	Chamber 3 LP - Connect Chamber 3 to Tank
u_8	Chamber 4 LP - Connect Chamber 4 to Tank

Table 5.3. Input Functions in regards to chamber pressures

5.2.2 States

The relevant states of the system that will be used during control are position of piston and its time derivative, along with pressures in the four chambers. For modelling, more external states are necessary, and these are supply pressure and load pressure, but are assumed to be constant in regards to the prediction model. Note that valve states are not part of the model. Valve states are removed to reduce complexity, but the valve dynamics are instead captured outside of the model which will be explained later in the prediction model section.

The system states are collected in $\boldsymbol{q}$.

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} x_p \\ \dot{x}_p \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

Constant Parameters

Constant parameters from the full model are reused, and expressions are evaluated to numerical values. The viscous damping is reduced to a coefficient as

$$B = \frac{A_s \mu_0}{h}$$

where the viscosity model is neglected. A_s is the whole contact area between piston and housing walls. This area is actually constant at all times, and is thus an accurate simplification. The damping term thereby considers the whole piston chamber system. Bulk modulus is assumed to be constant. Each chamber volume V is held constant. The value is found by evaluating a chamber volume, when the piston is at its middle position in the housing, which is the average chamber volume. The constants used in the prediction model are shown in table 5.4.

Model Constant	**Symbol**	**Value**	**Unit**
Piston Pressure Area	A_{set}	6.10e-5	m^2
Piston Mass	m_p	1.52	kg
Viscous Damping Coefficient	B	1.07e+2	Ns/m
Bulk Modulus	β_m	1.40e+9	Pa
Average Chamber Volumes	V	2.70e-5	m^3
Orifice Equation Coefficient	k_v	2.36e-7	$\frac{m^3}{s\sqrt{\frac{N}{m^2}}}$

Table 5.4. Reduced System Constants

Flows

Flows are calculated for every chamber for both on/off valves and check valves. Leakage is neglected. The flows are calculated in a simpler way than for the full model, since they primarily serve to simulate the pressures in the chambers. Every flow is assumed to have one function and one direction. This means that e.g. flow in regards to connection between chamber and HP manifold only flows into the chamber and not out. If the connection is ON and the chamber pressure surpasses the manifold pressure, then there will be outgoing flow via the check valve flow. This is regarded as a reasonable simplification since the only real difference between this and the full model is the distribution of flow between on/off valves and check valves, considering that the check valves are not significantly restrictive.

Inflow to chambers due to ON/OFF valve connected to manifolds

$$
\begin{aligned}
Q_{1,in} &= z_1 k_v \sqrt{|p_S - q_3|} \cdot (q_3 < p_S) \\
Q_{2,in} &= z_2 k_v \sqrt{|p_S - q_4|} \cdot (q_4 < p_S) \\
Q_{3,in} &= z_3 k_v \sqrt{|p_L - q_5|} \cdot (q_5 < p_L) \\
Q_{4,in} &= z_4 k_v \sqrt{|p_L - q_6|} \cdot (q_6 < p_L)
\end{aligned}
\tag{5.1}
$$

Outflow from chambers due to ON/OFF valve connected to tank

$$
\begin{aligned}
Q_{1,out} &= z_5 k_v \sqrt{|p_0 - q_3|} \cdot (q_3 > p_T) \\
Q_{2,out} &= z_6 k_v \sqrt{|p_0 - q_4|} \cdot (q_4 > p_T) \\
Q_{3,out} &= z_7 k_v \sqrt{|p_0 - q_5|} \cdot (q_5 > p_T) \\
Q_{4,out} &= z_8 k_v \sqrt{|p_0 - q_6|} \cdot (q_6 > p_T);
\end{aligned}
\tag{5.2}
$$

Inflow to chambers due to CHECK valves connected to tank

$$
\begin{aligned}
Q_{1,cv,in} &= k_v \sqrt{|p_T - q_3|} \cdot (q_3 < p_T) \\
Q_{2,cv,in} &= k_v \sqrt{|p_T - q_4|} \cdot (q_4 < p_T) \\
Q_{3,cv,in} &= k_v \sqrt{|p_T - q_5|} \cdot (q_5 < p_T) \\
Q_{4,cv,in} &= k_v \sqrt{|p_T - q_6|} \cdot (q_6 < p_T)
\end{aligned}
\tag{5.3}
$$

Outflow from chambers due CHECK valve connected to manifold

$$
\begin{aligned}
Q_{1,cv,out} &= k_v \sqrt{|p_T - q_3|} \cdot (q_3 > p_S) \\
Q_{2,cv,out} &= k_v \sqrt{|p_T - q_4|} \cdot (q_4 > p_S) \\
Q_{3,cv,out} &= k_v \sqrt{|p_T - q_5|} \cdot (q_5 > p_L) \\
Q_{4,cv,out} &= k_v \sqrt{|p_T - q_6|} \cdot (q_6 > p_L);
\end{aligned}
\tag{5.4}
$$

Differential Equations

The differential equations of the system are collected in $\dot{\boldsymbol{q}}$. First entry is piston speed, second is piston acceleration from Newton's second law, and entry 3-6 are continuity equations for each chamber respectively.

$$
\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} = \begin{bmatrix} \dot{x}_p \\ \ddot{x}_p \\ \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \\ \dot{p}_4 \end{bmatrix} = \begin{bmatrix} q_2 \\ \frac{1}{m_p} A_{set}(q_3 - q_4 + q_5 - q_6) - B q_2 \\ \frac{\beta_m}{V}(Q_{1,in} - Q_{1,out} + Q_{1,cv,in} - Q_{1,cv,out} - A_{set} q_2) \\ \frac{\beta_m}{V}(Q_{2,in} - Q_{2,out} + Q_{2,cv,in} - Q_{2,cv,out} + A_{set} q_2) \\ \frac{\beta_m}{V}(Q_{3,in} - Q_{3,out} + Q_{3,cv,in} - Q_{3,cv,out} + A_{set} q_2) \\ \frac{\beta_m}{V}(Q_{4,in} - Q_{4,out} + Q_{4,cv,in} - Q_{4,cv,out} - A_{set} q_2) \end{bmatrix}
\tag{5.5}
$$

This system of differential equations can be used for simulation.

5.2.3 Model Simulation

The simplest way of simulating differential equations is by forward Euler integration, which is used in the control scheme. There exist many other tools, some which are more efficient but require more work implementation wise. The simple method is deemed feasible as a starting method.

The procedure is as follows: Initial values are stated for time zero $\boldsymbol{q}_0$. The differential equations are evaluated at current time step to produce a gradient. This gradient is integrated over a single time step to produce the change in states using rectangular integration. This loops for the duration of the prediction horizon. A single step is evaluated as:

$$
\boldsymbol{q} = \boldsymbol{q}_0 + T_s \dot{\boldsymbol{q}} \tag{5.6}
$$

Sample Time

A sample time of $T_s = 0.1$ ms is chosen which is the same as the lab setup's sample time. Smaller sample times lead to more accurate simulations but at the cost of computational effort.

Switching Times to Model Inputs
As stated previously, the inputs to the system is a 8x1 vector $\boldsymbol{u}$, but this only for a single instant in time. To accommodate the whole prediction horizon, the switching times are converted to an input matrix $\boldsymbol{U}$ size $k_p\ X8$ i.e. length of prediction horizon time steps and width of no. of inputs. The input matrix is initialized as zeros for columns 1-4 and ones for columns 5-8, which means that HP input signals are off as default and LP input signals are on as default in correspondence with table 5.1. When using the input matrix, the simulation tool simply uses the row of inputs corresponding to the correct point in time.

Simulating Valve Delay
There are no valve models in the prediction model, but instead the effect of the valve dynamics are captured when simulating. The primary issue with the valves is the lag they introduce to the system. To capture this, a lag factor t_{lag} is introduced which is the approximate time from a valve input signal is given to flow occurring into a chamber. The lag factor is converted to time steps k_{lag}, and the input to the simulation at every time step is thereby delayed by this amount of time steps. This procedure requires initial values as many steps into the past as the length of the lag. A delayed valve input $z_{v,k}$ at a time step k can then be defined as the valve input signal delayed:

$$z_{v,k} = u_{v,k-k_{lag}} \tag{5.7}$$

5.3 Optimization Structure

Optimization is at the heart of MPC, and this sections aims to elaborate the optimization scheme. Optimization background throughout the rest of this section is based on Arora (2016).

The optimization problem is formulated as:

$$\begin{aligned} \min_{\boldsymbol{t}_{des}} \quad & J(\boldsymbol{t}_{des}) \\ \text{s.t.} \quad & \begin{cases} \boldsymbol{c_{ineq}}(\boldsymbol{t}_{des}) & \leq \boldsymbol{0} \\ \boldsymbol{A}_{ineq}\boldsymbol{t}_{des} - b_{ineq} & \leq 0 \\ \boldsymbol{b}_l - \boldsymbol{t}_{des} & \leq \boldsymbol{0} \\ \boldsymbol{t}_{des} - \boldsymbol{b}_u & \leq \boldsymbol{0} \end{cases} \end{aligned} \tag{5.8}$$

Where $J(\boldsymbol{t}_{des})$ is the scalar objective function. Minimizing this leads to the piston following its reference. The objective function is implicitly a function of the design variables collected in $\boldsymbol{t}_{des}$, which are selected switching times. The minimization is subject to several constraints. First entry contains nonlinear inequality constraints $\boldsymbol{c_{ineq}}(\boldsymbol{t}_{des})$ which implicitly are a function of design variables, which both ensures that the piston stays within positional bounds while evaluating whether the braking pressure timing is satisfactory. Second entry is a linear inequality constraint which ensures no overlap between design variables. The two last entries ensure that design variables are bounded by the time span of the prediction horizon. All terms in the optimization problem are elaborated through the rest of this section.

To solve the optimization problem, Matlab's general purpose fmincon is utilized. An in-depth exploration of finding the best optimization algorithm is not within the scope of the project and optimization options will not be further elaborated, but can be seen in the Matlab documentation in appendix E. Optimization settings are primarily bases on documentation of The MathWorks Inc. (2021)

5.3.1 Design Variables

The design variable vector has two elements defined as:

$$\boldsymbol{t}_{des} = \begin{bmatrix} t_{des,1} & t_{des,2} \end{bmatrix}^T = \begin{bmatrix} t_{I,HP,off} & t_{B,HP,on} \end{bmatrix}^T \tag{5.9}$$

The two entries in the design variable vector each define certain switching times, respectively the input HP turn off time, and the braking HP turn on time. With these two times, the remaining 14 switching times can be determined, which is a design choice motivated by the desire for simplifying the optimization problem while exploiting knowledge about how the switching times generally should be composed for desired system performance. The following decisions are therefore not conclusive, but simply a qualified proposal. The decision scheme is shown in table 5.5.

Decision no.	Switching Time	Value
1	$t_{I,HP,on}$	0
2	$t_{I,HP,off}$	$t_{des,1}$
3	$t_{I,LP,off}$	0
4	$t_{I,LP,on}$	T_p
5	$t_{O,LP,off}$	$t_{des,1}$
6	$t_{O,LP,on}$	T_p
7	$t_{B,LP,off}$	$t_{des,2} - t_{delay}$
8	$t_{B,LP,on}$	T_p
9	$t_{B,HP,on}$	$t_{des,2}$
10	$t_{B,HP,off}$	T_p

Table 5.5. Switching Times generation from Design variables

The logic behind the decisions for every switching time are:

1. The input pressure should turn on instantly. It should never turn on later during a half-cycle since the piston must be accelerated as soon as possible. Except if enough time has progressed and the input HP should turn off, but this is controlled by the first design variable.
2. Input HP turns off at the time of the first design variable.
3. When the input HP is turned on, the corresponding LP should be turned off, to avoid HP fluid going directly to tank.
4. The input LP turn on time is set to end of horizon. When the input HP is turned off, the chamber is sealed off, but since there will be piston movement, the chamber is decompressed, so the input pressure falls without the need to connect to tank. When the pressure reaches tank levels, fluid will flow into the chamber due to check valves.
5. The output LP turns off when the input HP turns off. This allows pressure to be built up in the output chamber by compression. The output pressure should be high exactly when - or slightly before - the input pressure becomes low, otherwise, if there were a time gap, some kinetic energy of the piston would simply be lost to friction, and the effective outflow per half-cycle would be reduced. On the other hand, the output pressure should not become high before turning off the input HP since that would not cohere with the

working principles where the input and output stage should be separated. If the output pressure becomes high when the input pressure is high, then some fluid is just pushed by the piston instead of transforming it by kinetic energy, and the piston might not reach high speeds. That can be a problem when transforming from a low input pressure to a higher output pressure.

6. Output LP turn on again at end of prediction horizon.
7. The braking LP should turn off when triggered by the second design variable. A small negative delay is included, such that the braking LP is turned off a short time before the braking HP is turned on. This allows pressure to be built up in braking chamber before opening to HP.
8. Braking HP turn on dictated by design variable 2.
9. Braking HP turn off when the prediction horizon is reached i.e. end of simulation.
10. Braking LP turn on again at end of prediction horizon.

For every half-cycle there is an unused chamber i.e. connected to tank, which removes 4 switching times, and it is never necessary to activate the output HP since the fluid will flow out of check valves. This removes 2 switching times, and there are 10 switching times left, solely defined by two design variables.

Anti Jitter

During a single half-cycle, an input should not turn on or off multiple times, although the optimization algorithm likes to do so if unhindered. This creates a jittery input signal, where an input turns on and off repeatedly for consecutive time steps, which is unwanted. Therefore, during every half-cycle the inputs connected with the design variables are monitored at every time step of simulating the plant. If an input changes, it triggers a flag, receiving a value of 1. If the input then changes again, it triggers a flag again, receiving a value of 2. An *if statement* then changes the switching times according to table 5.6. Both $u_{I,HP}$ and $u_{B,LP}$ are monitored and a triggered independently of each other. The principle is sketched in figure 5.5. It should be noted that the trigger is only evaluated in the plant simulation for the current time step i.e. the prediction horizon is not used. When the input signal is triggered, the input HP must turn off immediately, and the LP turns on. Meanwhile, the output LP is turned off immediately so pressure is kept. The kinetic potential of the piston must be used fully. If the braking pressure is triggered, then it must become HP as soon as possible, since the braking procedure already has started. This entails the LP to turn off.

Trigger 2: $u_{I,HP}$		**Trigger 2:** $u_{B,LP}$	
$t_{I,HP,off}$	$= 0$	$t_{B,HP,on}$	$= T_p$
$t_{I,LP,on}$	$= 0$	$t_{B,LP,off}$	$= T_p$
$t_{O,LP,off}$	$= 0$		

Table 5.6. Disable switching times if input has already switched

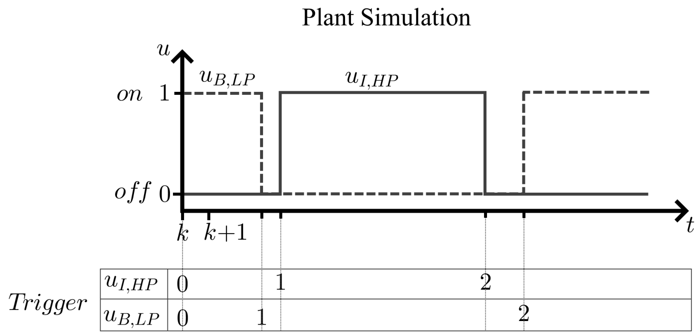


Figure 5.5. Input Jitter Trigger

5.3.2 Reference Position

The reference piston position changes dynamically depending on where the piston is, and what direction it is headed in. The reference can be two different positions; *high* or *low* which corresponds respectively to $x_{ref} = 5\,\text{mm}$ or $x_{ref} = 30\,\text{mm}$. If the piston reaches or has surpassed the *high* reference, then the reference should change to *low*. But if the piston changes direction during the simulation, it should not try to pursue the same reference. This means that if the sign of the velocity changes from one time step to another while the piston is above/below the middle position of 17.5 mm, then the reference changes. The reference position is updated based on table 5.7, where if **either** of the two conditions are true, then the target is updated.

To evaluate the conditions, some book keeping is necessary. Time step k refers to the current time step, so the current piston position is referred to as $x_{p,k}$. The piston position at previous time step is then $x_{p,k-1}$, and the system also applies to the piston speed. It must be noted that this is for global time and the reference is never defined inside a prediction.

Condition 1	**Condition 2**	**Target**
$30\,\text{mm} \leq x_p$	$sign(\dot{x}_{p,k-1}) \neq sign(\dot{x}_{p,k})$ & $17.5\,\text{mm} \leq x_{p,k}$	*high*
$x_p \leq 5\,\text{mm}$	$sign(\dot{x}_{p,k-1}) \neq sign(\dot{x}_{p,k})$ & $x_{p,k} \leq 17.5\,\text{mm}$	*low*

Table 5.7. Decision Table for choosing reference position

5.3.3 Objective Function

The optimization algorithm aims to minimize the objective function which interchangeably is called the cost function. For this control scheme, the cost function is single-objective, although a multi-objective cost function has been explored in earlier design iterations. The multi-objective formulation however showed unsatisfactory results, and its formulation along with tuning can be found in appendix D. The cost function is implicitly a function of the design variables, but the notation is dropped through the rest of the cost description.

The purpose of the cost function is to minimize the deviation between the piston trajectory and the reference. The reference position is passed to the cost function along with other relevant parameters, and inside the cost function, a prediction is evaluated resulting in a state matrix $\boldsymbol{Q}$ containing states for the span of the prediction horizon. From this matrix, the piston trajectory $\boldsymbol{x_p}$ is extracted. The piston trajectory $\boldsymbol{x_p}$ is evaluated for its peak value. That is the maximum value if the reference is high and the minimum value if the reference is low. The time step at which the peak occurs is stored as k_{peak}, and the piston position at peak is $x_{p,k_{peak}}$. The cost can then be defined as a quadratic expression:

$$J = \frac{1}{(0.025)^2}(x_{p,k_{peak}} - x_{ref})^2 \tag{5.10}$$

The fractional term is used to normalize the cost based on the maximum error that could be experienced if the piston stays within the *high* and *low* positions. The quadratic term is necessary such that the cost is non-negative in all cases. Only the peak position is compared to the reference, and an alternative formulation is to compare the whole trajectory with the reference, which is also a viable strategy. Peak value evaluation can be more effective since the optimization scheme should not consider what happens after the peak has been reached within a single prediction horizon. On the other hand, it can result in a slower trajectory, since there is no cost enticement as to what speed or time the reference is reached which would be a natural part of evaluating the whole trajectory. The peak evaluation method has the added benefit, that the piston speed is zero at the peak, which is a requirement for the piston to be able to commence the next half-cycle.

5.3.4 Constraints

There are several constraints for the optimization problem which include two nonlinear constraints: 1. piston position must be within bounds, such that there will not be a mechanical failure of the FBoT and 2. brake pressure timing to ensure proper precondition for following half-cycle. Furthermore there is a linear inequality constraint to ensure that there is no overlap between input and braking pressure. Finally there are lower and upper bounds of the design variables.

Linear Inequality Constraint

The linear inequality constraint is used to ensure that input HP and braking HP is not active at the same time. This means that the input HP signal must turn off before the braking HP signal turns on i.e. $t_{I,HP,off} < t_{B,HP,on}$. Rewritten in matrix form as

$$\boldsymbol{A}_{ineq}\boldsymbol{t}_{des} \leq \boldsymbol{b}_{ineq} \Rightarrow \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} t_{I,HP,off} \\ t_{B,HP,on} \end{bmatrix} \leq \begin{bmatrix} 0 \end{bmatrix} \tag{5.11}$$

Positional Constraint

This constraint ensures that the piston never exceeds its positional extremities, which otherwise could result in significant mechanical failure. The prediction model is evaluated and the trajectory is stored as $\boldsymbol{x}_p$ which contains piston positions for each step of the prediction horizon. A piston position at a certain time step k_1 is accessed as x_{p,k_1}. The trajectory must be confined within bounds as

$$1\,\text{mm} < x_{p,i} < 34\,\text{mm} \quad \text{for all } i \text{ from } 1 \text{ to } k_p \tag{5.12}$$

For every time step where this statement is not true, the absolute difference is stored in vector $\boldsymbol{Q}_{error}$. The constraint term is evaluated as the sum of elements in $\boldsymbol{Q}_{error}$, that is

$$c_Q = \sum_{i=1}^{k_p} Q_{error,i} \tag{5.13}$$

c_Q is hereby a scalar, being zero or a positive value. When it is equal to zero, there are no constraint violations in regards to piston position.

Braking Pressure Timing

The braking pressure should be high when the positional peak is reached, such that the pressure is high when the next half-cycle begins. This will be ensured with the following constraint formulation.

The results from the prediction model evaluation from the positional constraints are reused by extracting the chamber pressures. The braking pressure for the current mode and target reference is identified and defined as vector $\boldsymbol{p}_B$. The time steps where the pressure is high are identified and stored in a vector $\boldsymbol{v}_{pBidx}$, where an entry i is accessed as $v_{pBidx,i}$. The subscript refers to braking pressure index. The braking pressure should become HP just at the peak, so constraint violation is evaluated by whether the following statement is true:

$$v_{pBidx,1} \neq k_{peak} \tag{5.14}$$

If the statement is true, then the first time step of braking HP is not the same as the peak moment. If the statement is false, then the constraint value is zero i.e. no violation. If it is true, then it is beneficial to evaluate to what degree it is violated, which can be used to formulate a constraint gradient. If the constraint is violated, then the constraint value c_B is the absolute difference between moment of first braking HP and the peak:

$$c_B = \frac{1}{100}|v_{pBidx,1} - k_{peak}| \tag{5.15}$$

The fractional term is used to normalize the constraint value. It is not an exact normalization but serves to make the average constraint value, which is a sum of time steps, to be between zero and 1. As an example, 100 time steps means that the timing is off by 10 ms which will result in a constraint value of 1, which is assumed to be reasonable. It can be beneficial to define a constraint gradient, such the solver has information regarding how the design variables should be changed such that the constraint violation is eradicated. The braking HP turn off timing is naturally closely connected to when the braking pressure becomes high. In contrast to the objective function gradient, which will be elaborated in the subsequent subsection, the constraint gradient will not use finite difference approximations or other methods which requires multiple function evaluations. This is chosen to keep the computing time low. The gradient will instead be the constraint violation value with an appropriate sign. If the braking HP begins too soon, then the constraint gradient entry c_{Bg} will be assigned a negative sign. This means that when the design variable $t_{B,HP,on}$ is increased, the solver will interpret that the constraint violation will become smaller and vice versa. When the braking pressure becomes high too late, the opposite sign is applied with a similar logic. The constraint gradient must be a 2x2 matrix since there are two constraint terms and two design variables, but since the gradient should only concern the braking pressure timing, which only is influenced by the second design variable, only a single entry is non-zero. The constraint gradient matrix is then formulated as:

$$\boldsymbol{c}_{grad} = \begin{bmatrix} 0 & 0 \\ 0 & c_{Bg} \end{bmatrix} \tag{5.16}$$

The nonlinear constraints are collected in a vector and defined as inequality constraints as

$$\boldsymbol{c}_{ineq} < \boldsymbol{0} \tag{5.17}$$

where

$$\boldsymbol{c}_{ineq} = \begin{bmatrix} c_Q \\ c_B \end{bmatrix} \tag{5.18}$$

Lower and Upper Bounds
The design variables are bounded, such that the optimization does not try to use infeasible values. Design values cannot be less than 0 or larger than T_p represented respectively as a lower bound b_l and upper bound b_u. With two design variables, the bounds are defined as:

$$\boldsymbol{b}_l = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \boldsymbol{b}_u = \begin{bmatrix} T_p \\ T_p \end{bmatrix} \tag{5.19}$$

5.3.5 Objective Function Gradient Analysis

The optimization scheme uses sequential quadratic programming (sqp) to solve the optimization problem as a convex problem. To do this, the solver must have knowledge about the gradients of the cost function which must be programmed. Every evaluation of the cost function is relatively computationally costly, since a single evaluation requires a whole simulation of the prediction horizon. Therefore it is of interest to keep the number of evaluations as low as possible. Evaluating the gradient should result in a vector of gradients $\boldsymbol{c}$ corresponding to each design variable. The gradient is found by first evaluating the cost function with initial design variables. Then the first design variable is perturbed by a small amount t_{pert}, and the cost function is evaluated again. The difference between the two evaluated cost functions divided by the perturbation gives the gradient of the first design variable, which is the method of forward difference approximation. Some thought should be given to both the size of the perturbation and whether it is negative or positive. The perturbation should generally be small, so the local slope can be found, but not so small that effects cannot be seen. A positive perturbation means that the design value becomes larger i.e. a later time. This would have no effect on a variable which is already at the end of the prediction horizon since the variable would be perturbed beyond the upper bound. This would give a gradient of zero. Therefore, the first design variable is perturbed forward in time, and the second is perturbed backwards in time. The two gradient entries can then be formulated as:

$$c_1 = \frac{J(t_{des,1} + t_{pert}, t_{des,2}) - J(t_{des,1}, t_{des,2})}{t_{pert}} \tag{5.20}$$

$$c_2 = \frac{J(t_{des,1}, t_{des,2} - t_{pert}) - J(t_{des,1}, t_{des,2})}{t_{pert}} \tag{5.21}$$

and collected in $\boldsymbol{c}$ as:

$$\boldsymbol{c} = \begin{bmatrix} c_1 & c_2 \end{bmatrix}^T \tag{5.22}$$

5.3.6 MPC Schematic

The structure the MPC is shown in figure 5.7. Predictions are run at various stages in the scheme, and figure 5.6 shows the structure of a prediction simulation. For further details, see the Matlab appendix E which includes the numerous scripts needed to run the MPC.

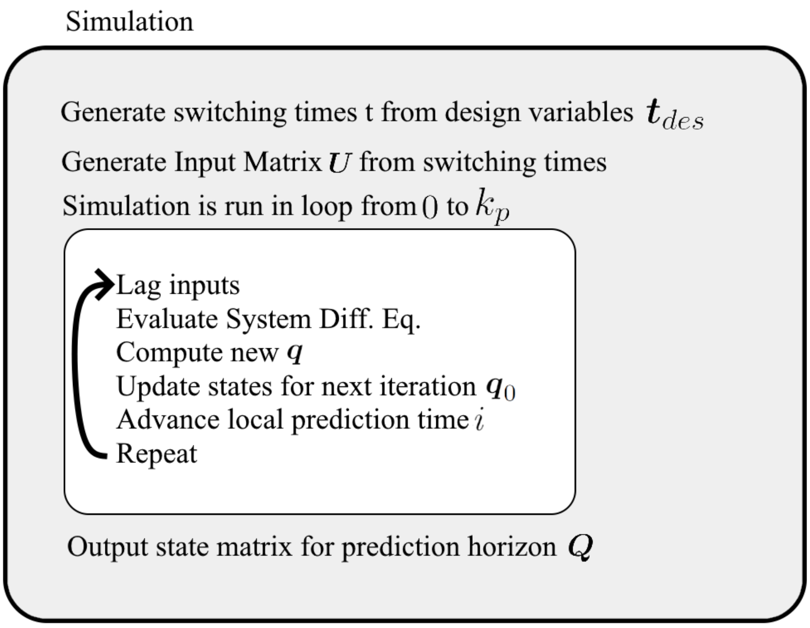


Figure 5.6. MPC Simulation function for prediction

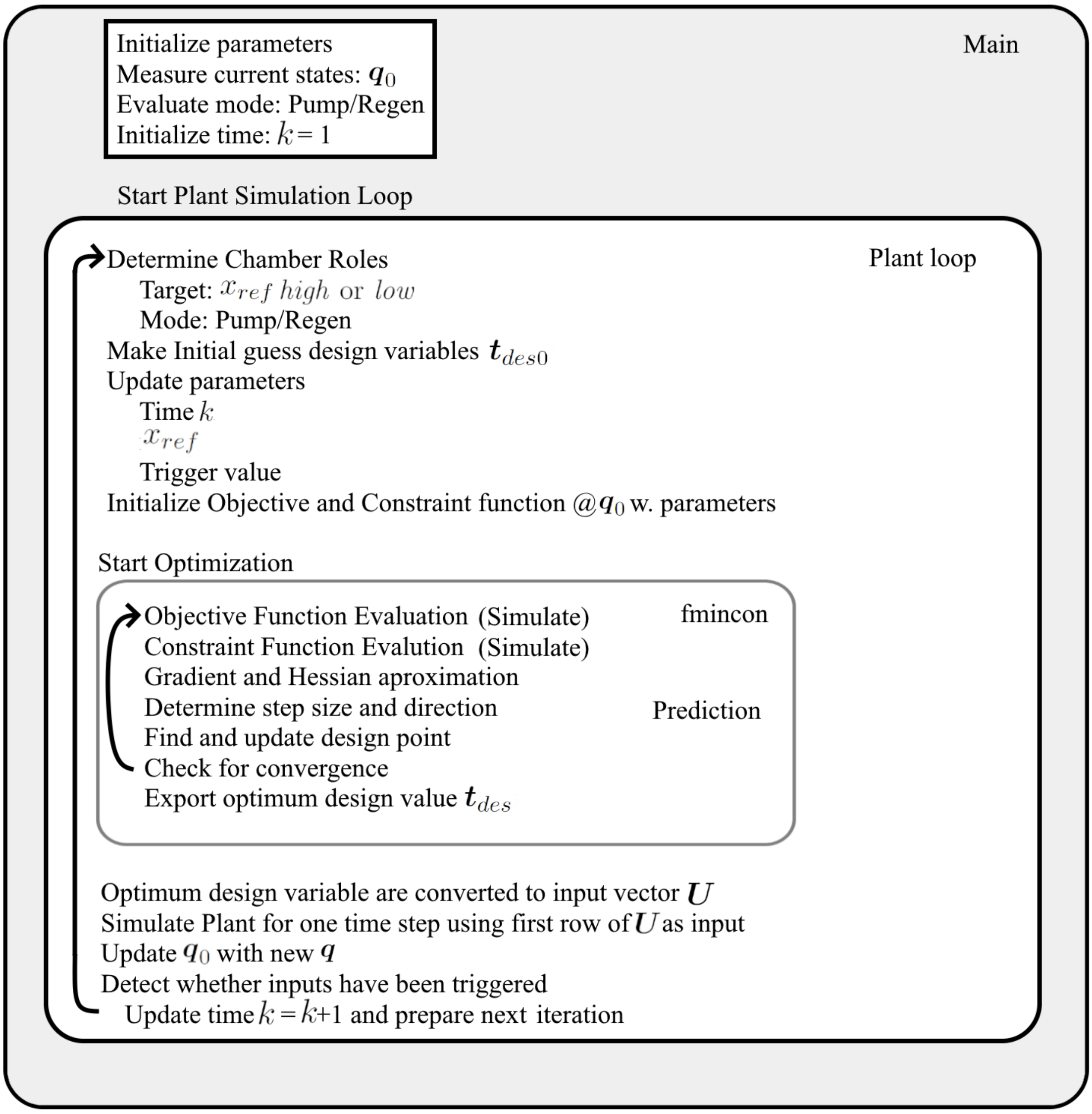


Figure 5.7. MPC Schematic

Chapter 6
MPC Evaluation

The performance of the control system will be evaluated by implementing the MPC with a virtual plant, and the results will be shown and discussed in this chapter. The virtual plant uses the same model as the prediction model, and is simulated in the same fashion. This corresponds to having a perfect model to represent a physical system, which practically is impossible. When implementing the control on a physical system, it will presumably be necessary to introduce an observer to estimate parameters that can improve the fidelity of the prediction model compared to the physical system. This will not be considered further in this chapter, which instead will focus on evaluation of the control system it self.

6.1 Demonstration

6.1.1 Pumping at Pressures Corresponding to Validation Data

This case is similar to that of the model validation. The supply and load pressures are set to $p_S = 80\,\text{bar}$ and $p_L = 50\,\text{bar}$. The mode is set to pumping and a graph for it running for 100 ms is seen in figure 6.1. The prediction horizon is set to 20 ms at a sample time of 0.1 ms.

The top part shows the piston position for the simulation time along with the reference which is generated dynamically. It is seen that the position stays within the positional limits, and reaches the reference perfectly and immediately changes the reference value. The frequency of the piston is $\approx 25\,\text{Hz}$ which is similar to the piston frequency from the model validation. The next row of the figure shows the chamber pressures. At first chamber 1 acts as the input chamber, and the initial condition of the chamber is set to the supply pressure. The high pressure accelerates the piston until $\approx 13\,\text{ms}$. At this point in time, the input pressure drops, and chamber 4 pressure rises and acts as the output. Until $\approx 20\,\text{ms}$ the piston is decelerated and chamber 2 pressure rises and first acts as braking pressure, and when the position reference switches, chamber 2 acts as input and the next half-cycle is begun. Chamber 2 pressure should be high before the piston reaches its peak, but that is not quite the case for the first half-cycle since p_2 is a little delayed. The bottom four rows in the figure show all valve input signals u. Solid colored lines show valves connecting to high pressure manifolds, and colored dashed lines show connection to tank. Each graph correspond to its own chamber. The lagged inputs z are shown in black and are equal to the inputs, except that they are delayed 2 ms. It can be seen that the delayed valve signals line up with chamber pressure changes chronologically. At first the chamber 1 valve input combination lets HP fluid into the chamber $u_1 = 1$, while connection to tank is closed $u_5 = 0$. However, the delayed signals does not carry past initial values, so both the HP input z_1 and tank connection z_5 is off which means the chamber effectively is sealed off. Since there already is HP in chamber 1, the piston accelerates, which expands the chamber volume. This decompresses the fluid which can be seen for the first $\approx 2\,\text{ms}$. After this time, the delayed signals match the input signals from time zero, and HP connection is reestablished which makes p_1 rise again. At $\approx 12\,\text{ms}$ chamber 1 is sealed off and the pressure falls rapidly due to the piston being at its maximum speed for the half-cycle. At the same time the output chamber pressure p_4 rises just as rapidly since the chamber has been sealed off. At $\approx 18\,\text{ms}$ a small glitch occurs where the braking chamber valve signals seal off the chamber, but then shortly afterwards the tank connection is established again.

This has the effect that p_2 rises shortly at first but then falls just as the next half-cycle is begun. It can be noted that p_2 rises at a slower rate than the previously mentioned p_1 and p_4 at the midpoint of the half-cycle. p_2 rises slower at the end of the half-cycle because the speed of the piston is slower. Ideally, p_2 should rise to the supply pressure solely due to chamber compression, but it has not been possible to achieve exactly that with the latest development of the control code. Instead, the high pressure valve u_2 is activated before the pressure has risen, which means there will be some losses due to a large pressure difference across the valve orifice. Disregarding this bug, the control scheme works as expected. A few other pressure and mode cases will be demonstrated as well.

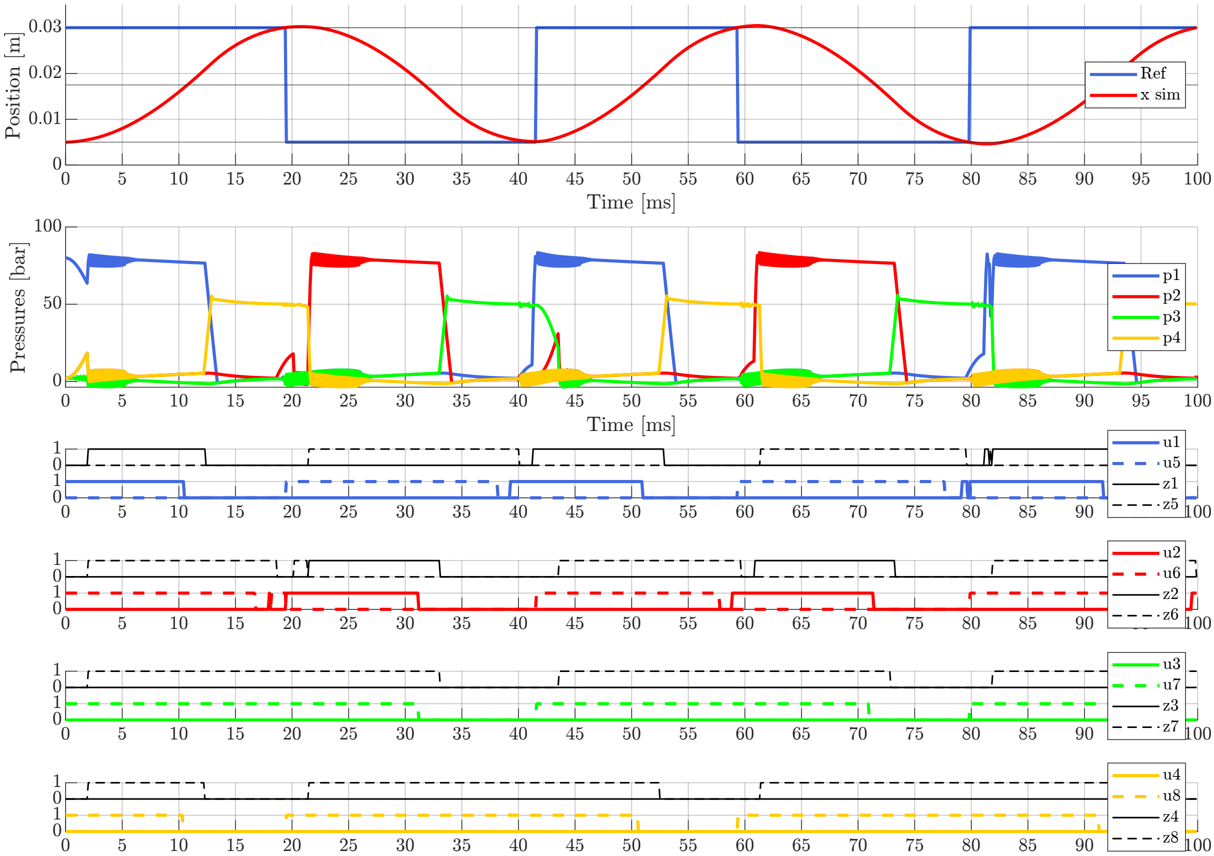


Figure 6.1. MPC Test on virtual plant in pumping mode and manifold pressures at $p_S = 80\,\text{bar}$ and $p_L = 50\,\text{bar}$

6.1.2 Regeneration

Using the same pressure case as with the previous demonstration, but in regeneration mode, the following results are found which are shown in figure 6.2. It can be seen that the position of the piston stays within the positional limits, but the frequency $\approx 20\,\text{Hz}$ is slightly lower compared to the pumping case. Since the input pressure is lower that the output, the input chamber must be pressurized for a longer time to transform enough energy to the piston before it being absorbed by the output chamber. This is exactly in thread with the working principle of the FBoT where the ratio of input pressure vs. output pressure ideally is equal to the ratio of output vs. input displaced volume. There are however losses which means that the theoretical input/output ratio is not entirely representative, which will be elaborated in section 6.2. The regeneration mode functions just as well as the pumping mode, but the losses during input pressure seem to have a significant effect on the performance, which will be explored in a more severe case in the following subsection.

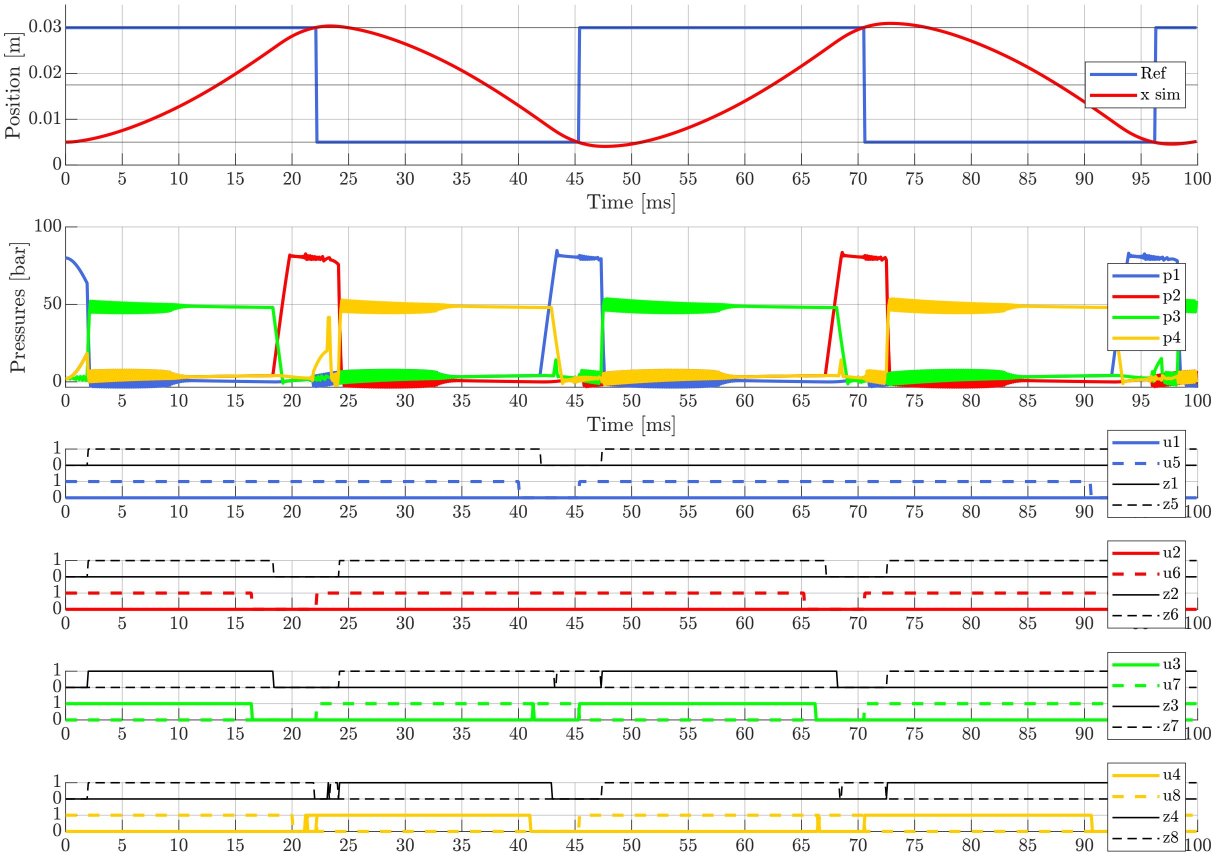


Figure 6.2. MPC Test on virtual plant in regenerate mode and manifold pressures at $p_S = 80\,\text{bar}$ and $p_L = 50\,\text{bar}$

6.1.3 Regeneration at Low Load Pressure

If the input pressure is too low, the FBoT will not function as intended which can be seen from this case. The supply and load pressures are set to $p_S = 100\,\text{bar}$ and $p_L = 30\,\text{bar}$ in regeneration mode, and the results are seen in figure 6.3. The low input pressure is unable to accelerate the piston sufficiently. The opposing viscous force seems to reach an equilibrium with the force from the input pressure, and a constant speed is reached. Near the reference position, the output chamber is activated, but since the output pressure is too high, the piston is quickly decelerated before the output pressure reaches its manifold pressure level. This means that there will be zero output flow, and all input energy is lost. Therefore, the load pressure needs to be sufficiently high before it is possible to regenerate fluid power.

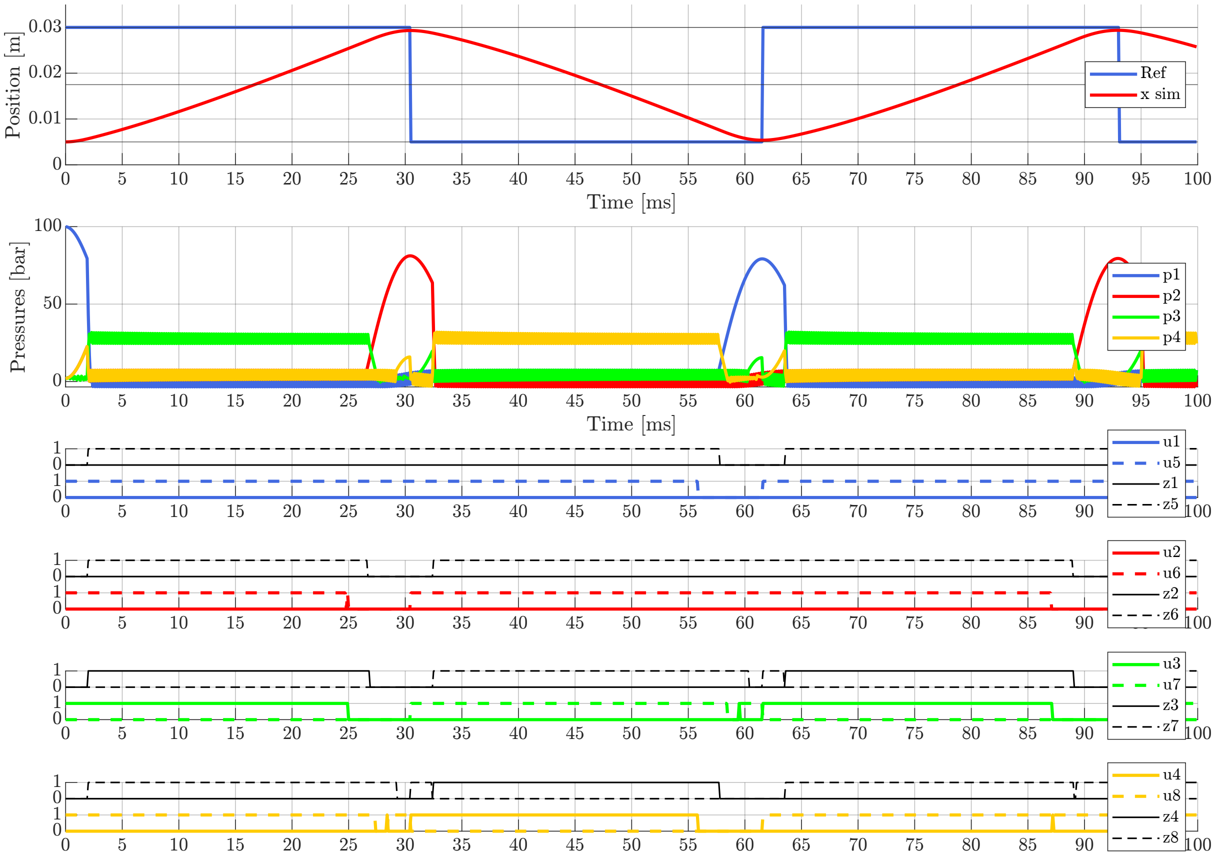


Figure 6.3. MPC Test on virtual plant in regenerate mode and manifold pressures at $p_S = 100\,\text{bar}$ and $p_L = 30\,\text{bar}$

6.1.4 Equal Manifold Pressures

The case of equal manifold pressures is demonstrated in pumping mode. The supply and load pressures are set to $p_S = 100\,\text{bar}$ and $p_L = 100\,\text{bar}$. The results are seen in figure 6.4. This case visualizes the inherent losses due to viscous damping by the fact that the time spent with high input pressure is longer than the time spent on high output pressure, which ideally should be equal. However, at a first glance, the output time is approximately half that of the input time, which indicates an approximate efficiency of 50%. The efficiency will be elaborated in the subsequent section, but despite the losses, the FBoT is still capable of proper transformation. Glitches still occur as sudden pressure spikes, which indicate that further work is needed before implementation on the lab setup.

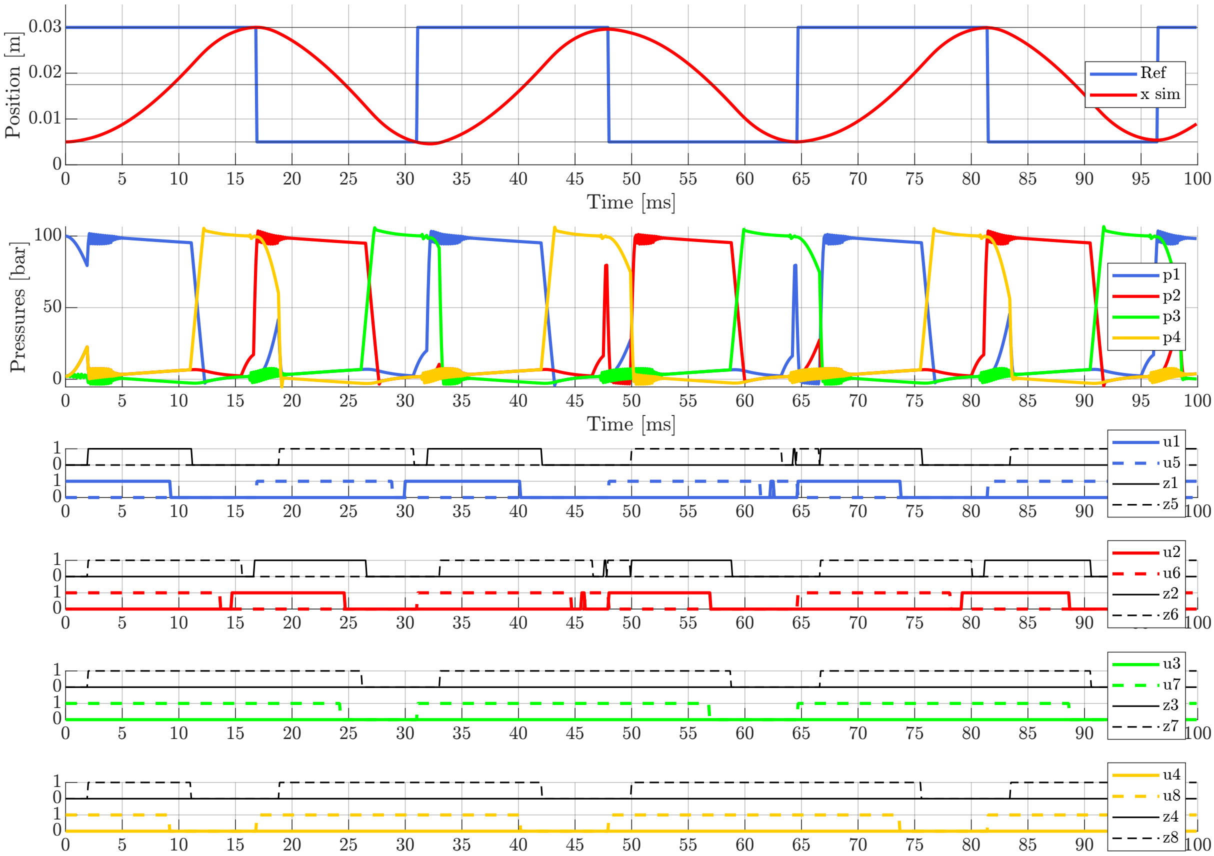


Figure 6.4. MPC Test on virtual plant in pumping mode and manifold pressures at $p_S = 100\,\text{bar}$ and $p_L = 100\,\text{bar}$

6.2 Efficiency

There are losses in the system, and this can be seen by the fact that the ratio of time that input and output pressure is high is different than 1 in the case of equal pressures. There is put more energy into the system than is extracted from it. This happens since energy is lost to friction, so the input energy must be sufficiently larger c.f. a basic energy consideration about work in vs. work out.

$$W_I = W_O + W_{loss} \tag{6.1}$$

To evaluate how efficient the control system for the FBoT is, there will be made an evaluation of work done on the piston by the input pressure vs. work done on the piston by the output pressure. A general equation for work W done by a force F tangentially along a curve s can be written as:

$$W = \int F ds \tag{6.2}$$

$$\tag{6.3}$$

The pressure force in all chambers always act tangentially along the piston direction, but the pressure forces change during the motion in an unpredictable way. To evaluate the work in the discrete time simulation, it is practical to integrate partial work over each time step denoted as δW_k and summing all terms to evaluate total work. The force from chamber pressures at each time step is denoted as F_k.

$$\delta W_k = F_k \int_{x_{p,k}}^{x_{p,k+1}} ds = F_k(x_{p,k+1} - x_{p,k}) \tag{6.4}$$

$$W = \sum_{k=1}^{k_{end}} \delta W_k \tag{6.5}$$

Expanding to include input and output pressures, the partial work equation for input and output work becomes

$$\delta W_{I,k} = A_{set}(p_{1,k} - p_{2,k})(x_{p,k+1} - x_{p,k}) \tag{6.6}$$

$$\delta W_{O,k} = A_{set}(-p_{3,k} + p_{4,k})(x_{p,k+1} - x_{p,k}) \tag{6.7}$$

Summing all terms for a plant simulation time of 1 s in pumping mode, with settings as described in section 6.1.1, reveals the total work and efficiency as:

$$\eta = \frac{W_O}{W_I} \tag{6.8}$$

A screening of efficiencies is conducted for different pressure combinations in both pump and supply mode, and the results are shown in table 6.1. The first pressure combination represents that from the model validation and the firstly demonstrated control data in the current chapter. The intention of the next column of pressure combinations show the impact of doubling the supply pressure and the last column show the efficiency for equal pressure levels. It is important to note that the efficiency evaluation does not consider valve losses, and it does not evaluate effective flows, but rather the amount of work the pressures exerts on the piston. As seen in the table, the efficiency of all cases is near 50% but slightly different in each case. This can be due to the aforementioned pressure glitches, but the results can also give the indication that higher pressures can yield higher efficiencies. The efficiency for regeneration mode at pressures $p_S = 160\,\text{bar}$, $p_L = 50\,\text{bar}$ is set to zero since the output pressure never reach the manifold pressure i.e. the FBoT does not function as intended.

	Manifold Pressure Combinations		
Mode	$p_S = 80\,\text{bar}$ $p_L = 50\,\text{bar}$	$p_S = 160\,\text{bar}$ $p_L = 50\,\text{bar}$	$p_S = 100\,\text{bar}$ $p_L = 100\,\text{bar}$
Pump	$\eta = 0.44$	$\eta = 0.47$	$\eta = 0.50$
Regenerate	$\eta = 0.41$	$\eta = 0.00$	$\eta = 0.51$

Table 6.1. Efficiency Screening Table

6.3 Tuning of Prediction Horizon

The main computational effort of the MPC lies in the evaluation of the prediction model. If the length of the horizon can be shortened i.e. reduce number of time steps k_p, then a control step can be evaluated faster. There may however not be a single best horizon since the piston frequency depends on the manifold pressure combinations. At first, one could think that the most important aspect of the prediction horizon is that it can encompass an entire half-cycle. This is however not the case which will be shown. For the case shown in figure fig. 6.1 on page 43, a half-cycle lasts $\approx 20\,\text{ms}$ and the prediction horizon is set to the same length. The prediction horizon at the very beginning of the case is shown in figure 6.5. A prediction horizon of 15 ms is shown in figure 6.6, where it can be seen that the end of the half-cycle is not encompassed. The prediction does not utilize the output pressure at first, but that does not necessarily mean that the horizon length is inadequate. The MPC consistently uses the first input signal on the plant and discards the rest before making a new prediction, and at first, the input signal opens the input HP which agrees with the prediction model with a longer horizon. After some time has passed, the prediction model with the short horizon will begin to utilize the output pressure, and ind the end, there may only be a small difference between control signals given to the plant. This statement is further supported by evaluating plant simulations for different prediction horizons as seen in figure 6.7. Meanwhile, shorter prediction horizons lead to shorter evaluation times per control step, and the corresponding correlations are stated in table 6.2. It can be seen from the results, that a horizon length of 10 ms is adequate for the given case while the evaluation time is significantly shorter compared to the initial horizon of 20 ms. This means that the prediction horizon can be shorter than a half-cycle without compromising performance while shortening the evaluation time. The optimal horizon length will depend on pressure combinations, sample time and valve dynamics.

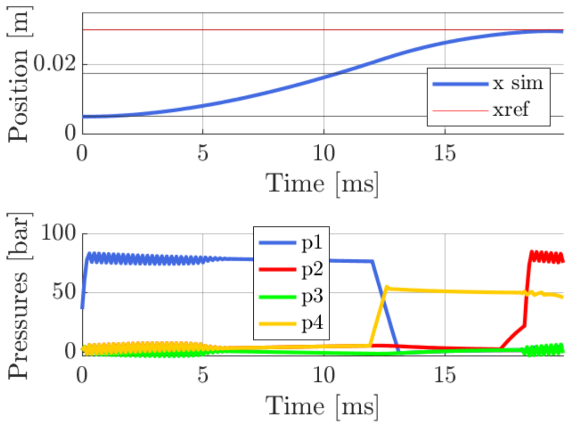


Figure 6.5. Prediction horizon of 20 ms

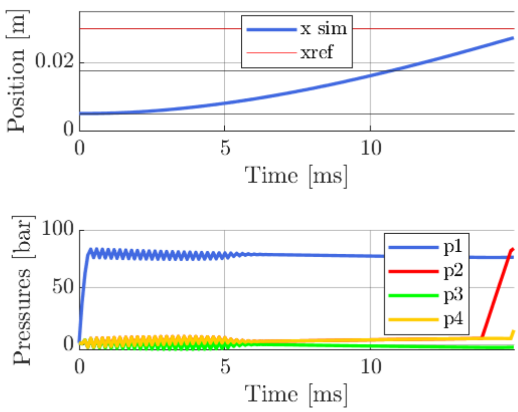


Figure 6.6. Prediction horizon of 15 ms

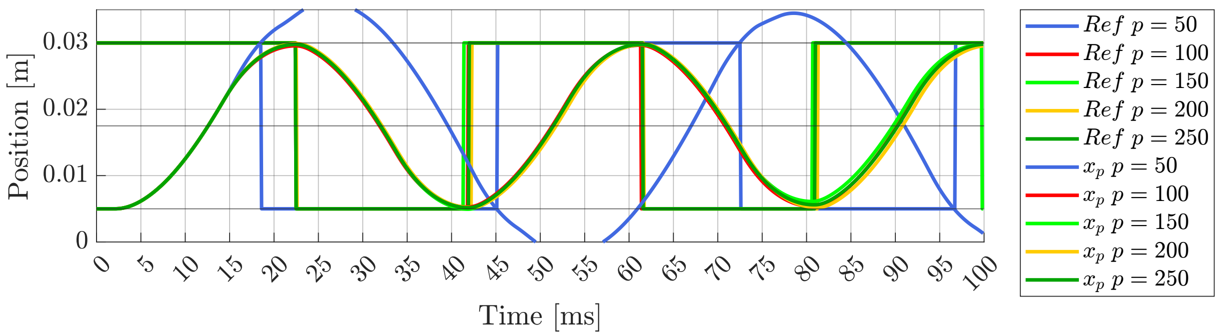


Figure 6.7. Horizon Tuning

Horizon Length	**25ms**	**20ms**	**15ms**	**10ms**	**5ms**
Evaluation Time	107ms	79ms	56ms	23ms	13ms

Table 6.2. Average evaluation time per control step for different prediction horizon lengths

This concludes the evaluation of the MPC implemented on a virtual plant.

Chapter 7
Discussion

7.1 Control Strategies

It became evident from the control development that there are numerous viable strategies for controlling the FBoT with MPC in more or less complicated fashions. In this report, it was sought to make the control problem as simple as possible to reduce computation time while increasing robustness. A more complicated problem, for example with all 16 switching times as design variables, require more constraints and facilitates different ways to minimize the objective function. But by considering the problem, it was found that several degrees of freedom could be removed such that the MPC only would solve the optimization problem in one way. This gave more consistent results and made it possible to use a single-objective cost function in contrast to multi-objective functions, which were necessary with a more design variables. This simplification however made it difficult to adhere to the specification that valves only must open when the pressure difference across it is sufficiently small, since the delay between closing off a chamber and opening for high pressure is fixed, but should depend on manifold pressures and piston speed.

One radical simplification that all control strategies can benefit from is the idea of assigning roles to each chamber based on whether the FBoT is in pumping mode or regeneration mode and whether direction the piston is moving. This simplification reduces the control problem as long as the control algorithm can change between these roles effortlessly.

7.2 State and Parameter Observer

The MPC was tested on a virtual plant with the exact same structure and parameters as the prediction model. This means that the predictions will be very similar to the simulated result of the virtual plant. However, this does not mean that the MPC is ready for implementation on the physical system because there will be differences between the prediction model and the physical system. Large infidelities will then essentially result in the MPC finding an optimal solution for the wrong problem, as the physical system will require different inputs because of different dynamics. It is therefore important to implement an observer which can estimate prediction model parameters from the physical system. In particular, such a parameter could be the viscous damping coefficient as this has a significant influence on how fast the piston will move which is important for when to switch input signals. Furthermore it can be beneficial to estimate valve dynamics.

7.3 Efficiency

The underlying purpose of the FBoT is to transform fluid power as efficiently as possible. Theoretically the FBoT is capable of this, but there are significant losses from viscous damping occurring when the piston moves. Evaluations of the control of the FBoT on a virtual plant showed a maximum efficiency of $\approx 50\%$, which is an optimistic estimation, since not all loss factors are included. However, this study is based on a fixed viscous damping coefficient which was found during system modelling and validated using a single physical experiment. The viscous damping is however not constant and is influenced by temperature in particular, which has not been a focus point in the system modelling.

For the intended working principles of the FBoT, the developed control system is regarded to be effective.

7.4 Constraints

For the FBoT to function as efficiently as possible, valves must only open when the pressure difference across them are sufficiently low, as previously stated. This could be seen as a constraint, and earlier design iterations included this as an optimization constraint. It was however not possible to abide by this constraint while achieving satisfactory piston movement. This is mainly because of the reduced amount of design variables which limits the freedom of the control problem and how switching times are defined from the design variables. The FBoT control therefore does not function as efficiently as intended, which potentially can be the objective of future design development. There is however a caveat which makes it less obvious whether the FBoT is more efficient with the intended valve timings in contrast to the current design where valves are opened between a low and high pressure. Consider how a chamber pressure rises due to compression. Taking offset in the continuity equation with zero flows:

$$\frac{V}{\beta}\frac{dp}{dt} = -\frac{dV}{dt} \tag{7.1}$$

Restructuring the equation and by separation of variables, the following integral equation is found

$$\int \frac{1}{\beta}\, dp = -\int \frac{1}{V}\, dV \tag{7.2}$$

Integrating from an initial pressure and volume p_0 and V_0 to a different pressure and volume p_1 and V_1, the integral equation can be evaluated and simplified to

$$V_0 \exp\left(\frac{p_0 - p_1}{\beta}\right) = V_1 \tag{7.3}$$

With this reformulation of the continuity equation, the necessary volume compression needed for some pressure difference can be found. Considering the case of pumping mode with a supply pressure $p_S = 150\,\text{bar}$ at the end of a half-cycle where the braking pressure should be built up by compression, the needed piston movement length can be found. There will be assumed an average bulk modulus of $\beta = 7500\,\text{bar}$ and initial chamber pressure $p_0 = 1\,\text{bar}$. Furthermore the the piston pressure area $A_{set} = 61\,\text{mm}^2$ and initial volume of $V_0 = 2.700 \times 10^{-5}\,\text{m}^3$ which is the average chamber volume used, and the compressed volume is then found by evaluating equation 7.3 as $V_1 = 2.647\,\text{m}^3$. The difference between volume V_0 and V_1 corresponds to moving the piston $\Delta x_p = 8.7\,\text{mm}$ which is a considerable part of the total allowable movement of the piston of 25 mm. This means that the chamber compression needs to start some time before the piston peak is reached, and the necessary length depends on the manifold pressure. Consider figure fig. 6.4 on page 46 where the supply pressure is $p_S = 100\,\text{bar}$. Here, the compression length would be $\Delta x_p = 5.8\,\text{mm}$, and if the braking pressure must be high at peak, then the compression must start at time $\approx 12\,\text{ms}$ which would overlap considerably with the output pressure i.e. a considerable amount of work is put into compressing the braking chamber instead of the output chamber. It must be considered that the bulk modulus has a an effect on the compression length, so the compression lengths can only be seen as approximate.

Chapter 8
Conclusion

In this project, the fluid power Full Bridge Oscillation Transformer (FBoT) developed at Aalborg University has been under consideration in regards to modelling and control. The FBoT concept would theoretically be able to transform fluid power at perfect efficiency by applying concepts of energy conversation to a mechanically simple system. The FBoT consists of four chambers surrounding a free-floating piston, and each chamber can be connected to either a supply or load manifold and the tank. The working principle of the FBoT is to let an input supply flow into a chamber of the FBoT and let the input pressure accelerate the free-floating piston at first. The kinetic energy of the piston is then transformed to output fluid power subsequently at a potentially different pressure and flow rate. The goal of the project has been to develop a control system to facilitate this working principle. To develop a control system, a model of the system was first needed. A detailed model has been developed and validated with data from the lab setup, where the FBoT is implemented in an electric wheel loader. With the model in place, control strategies could be developed, and it was soon realized that advanced control strategies were needed. The control strategy choice fell on Model Predictive Control (MPC) which uses a prediction model to be able to accommodate future system responses, and thereby plan a whole time series of inputs, such that the FBoT behaves as planned. This is in contrast to reactive controllers with defined control laws.

A unique MPC was developed for the FBoT which was made possible by optimizing valve switching times and reducing the control problem by assigning each chamber different roles dependent on whether the FBoT functions in pump mode or regeneration mode.

From the MPC evaluation in chapter 6, several of the requirements stated in chapter 4 could be answered. It has been possible to make the piston move reciprocally between two positional limits, smoothly for various cases. The input pressure is transformed to kinetic energy of the piston which in turn compresses fluid in the opposing load chamber which creates output flow. The combination of valve switching times ensure that there never is a direct connection between a high pressure manifold and the tank, but it has however not been possible to avoid losses in all cases that occur when opening a valve with a large pressure difference across. It is furthermore shown that the control works for various cases including pumping mode where supply power is transformed to the load, and regeneration mode where power is transformed from the load to the supply. It is possible to transform from high to low pressure as well as low to high pressure. This gives a robustness to the control system. There is however a limit on how low the input pressure can compared to the output pressure, because of energy losses due to viscous damping. Such a case does not result in mechanical failure, but there will simply be no output flow. The fact that there are constraints on position of the piston adds to the robustness. The control system has solely been tested on a virtual plant, and it is assumed that for physical setup implementation an observer is necessary to estimate states and parameters for the optimization model.

It has been shown, based on a virtual plant evaluation, that it is possible to develop a robust control system that facilitates the FBoT to function as intended by transforming fluid power.

Chapter 9
Reflections

9.1 Future Work

The MPC has been implemented on a virtual plant which showed promising results. Before using it on the lab setup, it will be reasonable to first implement the control on the Simulink model of the system which is more representative of the lab setup than the prediction model. Furthermore an observer, potentially a kalman filter, can be used to estimate states and parameters to repair the prediction model persistently.

The prediction model is severely reduced compared to the full model, and it might be beneficial to increase the complexity of the prediction model if the MPC is able to run fast enough.

Work on the optimization algorithm is also recommended since not all constraints are met at all time instances, and there may be better ways to decide the switching times based on the design variables.

9.2 Efficiency Evaluation

The efficiency has been estimated only using the reduced plant model, and does not consider actual flows, which would be a more comprehensive way to evaluate efficiency of the FBoT.

9.3 Lessen Damping

The damping in the FBoT is considerable. Other control strategies might be developed which moves the piston at lower frequency and with a smaller amplitude which would diminish losses due to viscous damping. This could as an example potentially be achieved with an objective function which aims to minimize losses, which naturally requires a fast way to evaluate losses.

Bibliography

Arora, 2016. Jasbir Arora. *Introduction to Optimum Design*. Elsevier Science Publishing Co Inc, 2016. ISBN 9780128008065.

García et al., 1989. Carlos E. García, David M. Prett and Manfred Morari. *Model Predictive Control: Theory and Practice a Survey*. Automatica, Vol. 25(No.3), pp. 335–348, 1989.

Hansen, 2023. Anders Hedegaard Hansen. *Fluid Power Systems*. Springer, 2023. ISBN 978-3-031-15089-0.

Johansen et al., 2015. Per Johansen, Daniel B. Roemer, Torben O. Andersen and Henrik C. Pedersen. *DELTA-SIGMA MODULATED DISPLACEMENT OF A DIGITAL FLUID POWER PUMP*. Proceedings of: The Seventh Workshop on Digital Fluid Power, February 26–27, 2015, Linz, Austria, pp. 1-9, 2015.

Knežević and Savić, 2006. Darko Knežević and Vladimir Savić. *MATHEMATICAL MODELING OF CHANGING OF DYNAMIC VISCOSITY, AS A FUNCTION OF TEMPERATURE AND PRESSURE, OF MINERAL OILS FOR HYDRAULIC SYSTEMS*. Mechanical Engineering, Vol 4(No.1), pp. 7–34, 2006.

The MathWorks Inc., 2021. The MathWorks Inc. *Optimization Toolbox™ User's Guide (v.R2021a)*, 2021. URL `https://www.mathworks.com`.

Vinderup Maskiner, 2023. Vinderup Maskiner. *1026 LX/C ELECTRIC*, 2023. URL `https://www.vmloader.dk/no/modeller/1026-lxc-electric.html`. Last visited: 30/05/2023.

Volvo CE, 2021. Volvo CE. *Groundbreaking electro-hydraulic system wins Volvo Technology Award*. Press Release, 2021. URL `https://www.volvoce.com/global/en/news-and-events/press-releases/2021/groundbreaking-electro-hydraulic-system-wins-volvo-technology-award/`.

Appendix A
Flows in Flow Diagram

Pressure control volumes:
p_S: Supply line manifold
p_L: Load Side manifold
p_A: Load chamber A
p_B: Load chamber B
p_{C1}: Piston Chamber 1 - Supply Side
p_{C2}: Piston Chamber 2 - Supply Side
p_{C3}: Piston Chamber 3 - Load Side
p_{C4}: Piston Chamber 4 - Load Side

Flow overview

Valve Flows excluding piston chambers
$Q_{V1T} = Q_{V1} - Q_{CV1}$ Valve 1 to tank
$Q_{V3T} = Q_{V3} - Q_{CV3}$ Valve 3 to tank
$Q_{V5T} = Q_{V5} - Q_{CV5}$ Valve 5 to tank
$Q_{V7T} = Q_{V7} - Q_{CV7}$ Valve 7 to tank

$Q_{V2L} = -Q_{V2} + Q_{CV2}$ Valve 2 to Load
$Q_{V8L} = -Q_{V8} + Q_{CV8}$ Valve 8 to Load

$Q_{S1} = Q_{V6} - Q_{CV6}$ Supply to Valve 6
$Q_{S2} = Q_{V4} - Q_{CV4}$ Supply to Valve 6

Q_{V1T}
Q_{V3T}
Q_{V5T}
Q_{V7T}
Q_{BT}
all go to tank

$Q_L = Q_{V2L} + Q_{V8L}$ Load flow into the load chamber
Following is inflow for the piston chambers
$Q_{C1} = -Q_{V5} + Q_{CV5} + Q_{V6} - Q_{CV6}$ Chamber 1 inflow
$Q_{C2} = -Q_{CV4} + Q_{V4} + Q_{CV3} - Q_{V3}$ Chamber 2 inflow
$Q_{C3} = -Q_{CV8} + Q_{V8} + Q_{CV7} - Q_{V7}$ Chamber 3 inflow
$Q_{C4} = -Q_{V1} + Q_{CV1} + Q_{V2} - Q_{CV2}$ Chamber 4 inflow

Piston Chamber leakage flows
$Q_{le,C1T}$ Leakage from chamber 1 to tank
$Q_{le,C1C3}$ Leakage from chamber 1 to chamber 3
$Q_{le,C3T}$ Leakage from chamber 3 to tank
$Q_{le,C2T}$ Leakage from chamber 2 to tank
$Q_{le,C2C4}$ Leakage from chamber 2 to chamber 4
$Q_{le,C4T}$ Leakage from chamber 4 to tank

Appendix B
Simulink Model

The implementation of the model in Matlab's Simulink is shown here in figures B.1 through B.10.

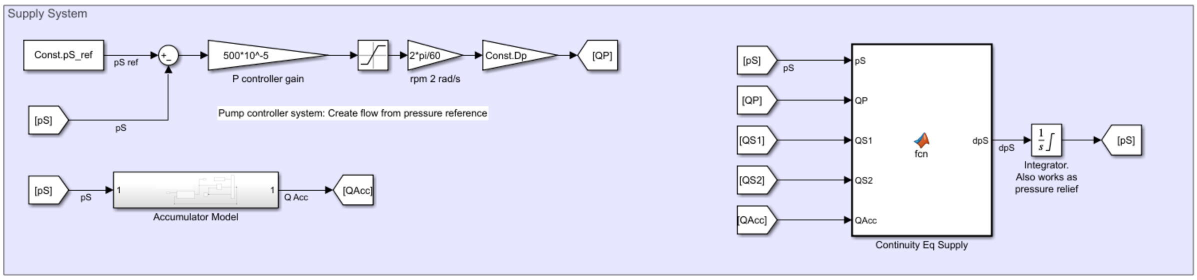


Figure B.1. Simulink Supply System

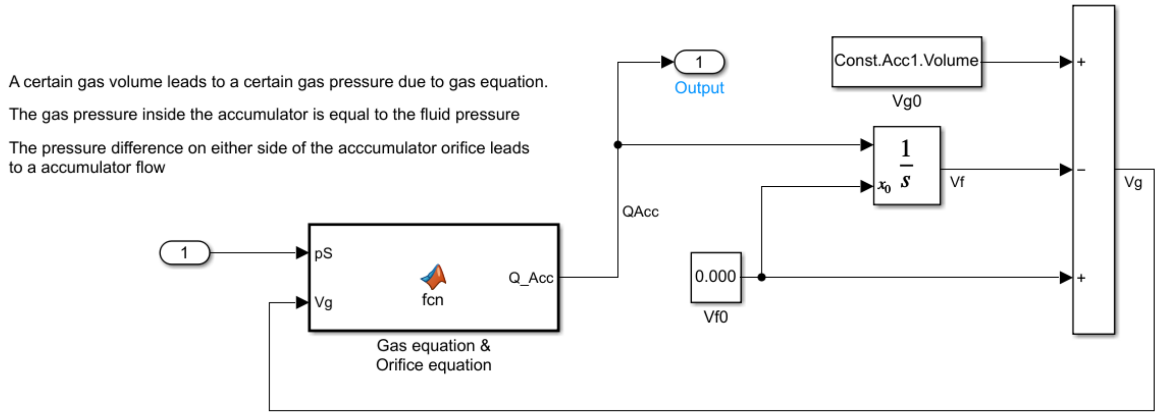


Figure B.2. Simulink Accumulator subsystem

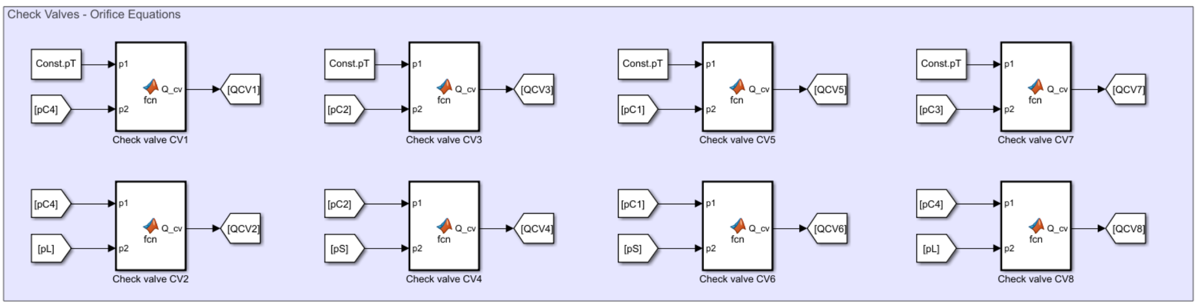


Figure B.3. Simulink Check Valves

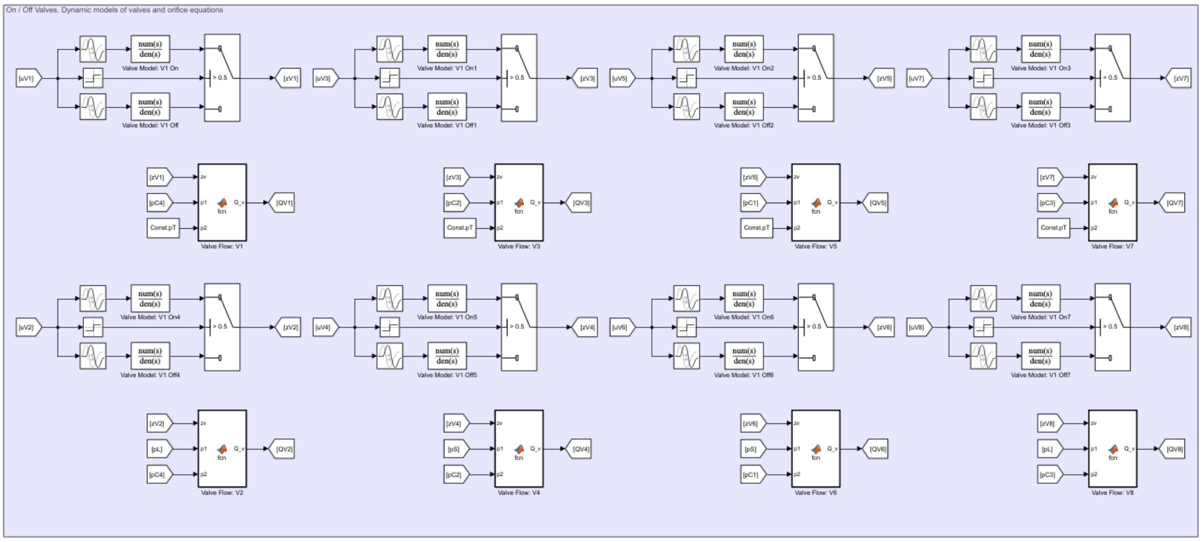


Figure B.4. Simulink On Off Valves overview

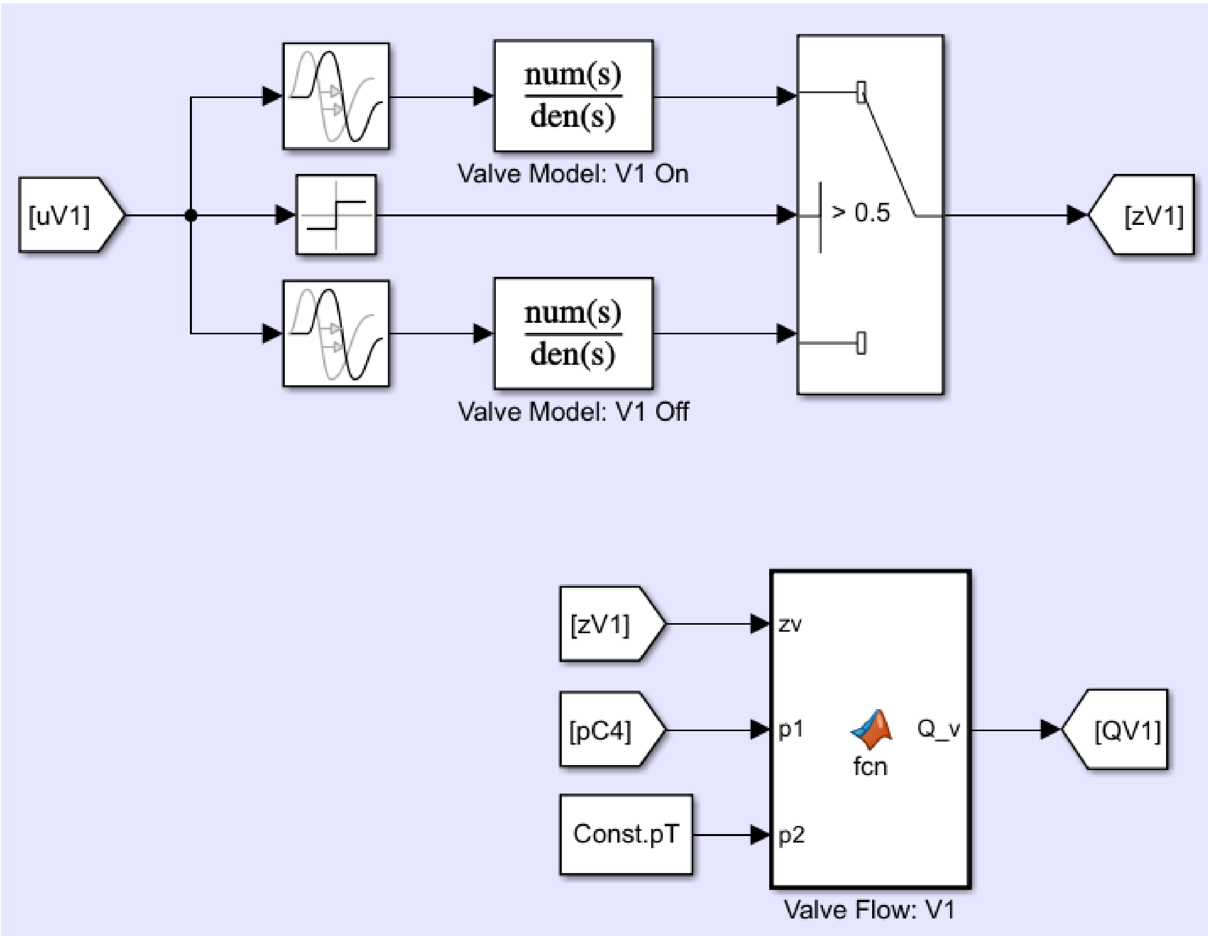


Figure B.5. Simulink On Off Valves Zoom-in on single valve

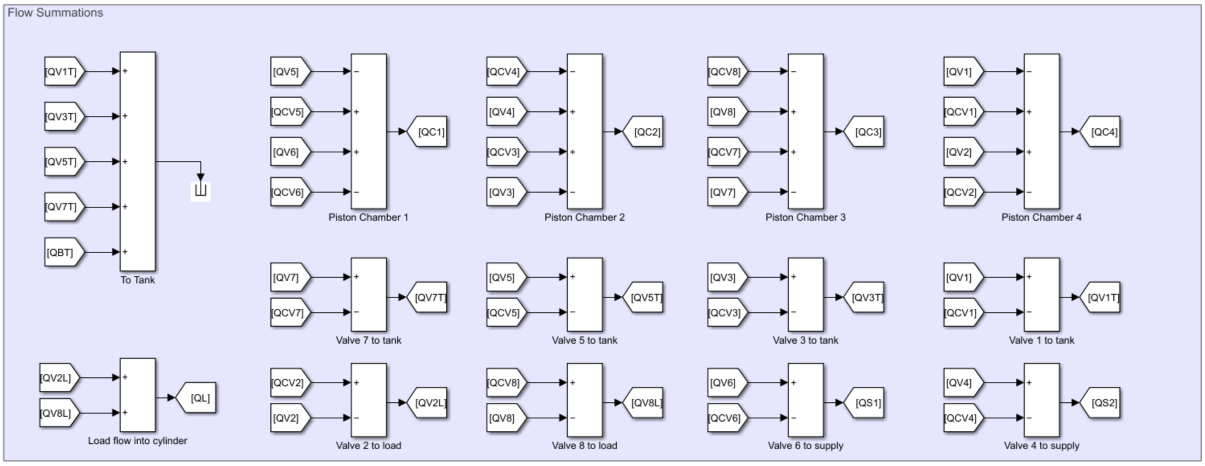


Figure B.6. Simulink flow summations

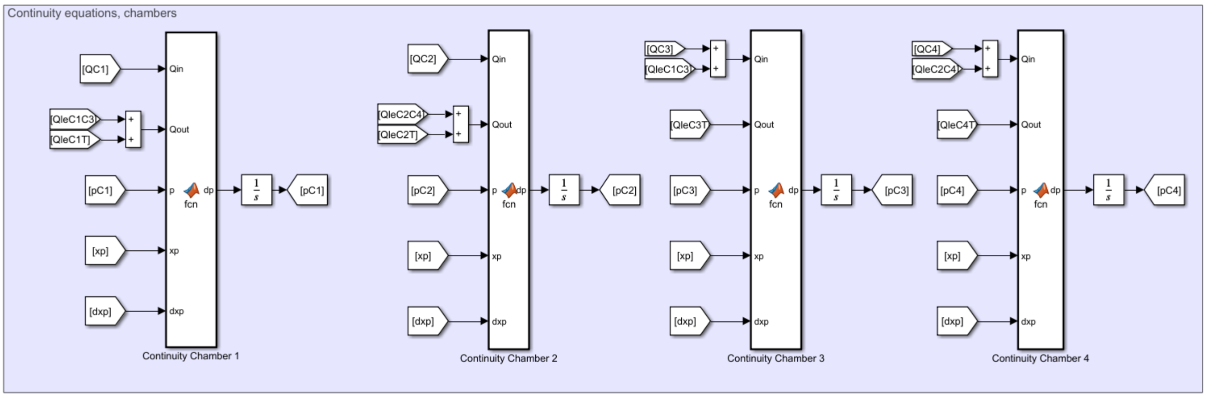


Figure B.7. Simulink Continuity equations chambers

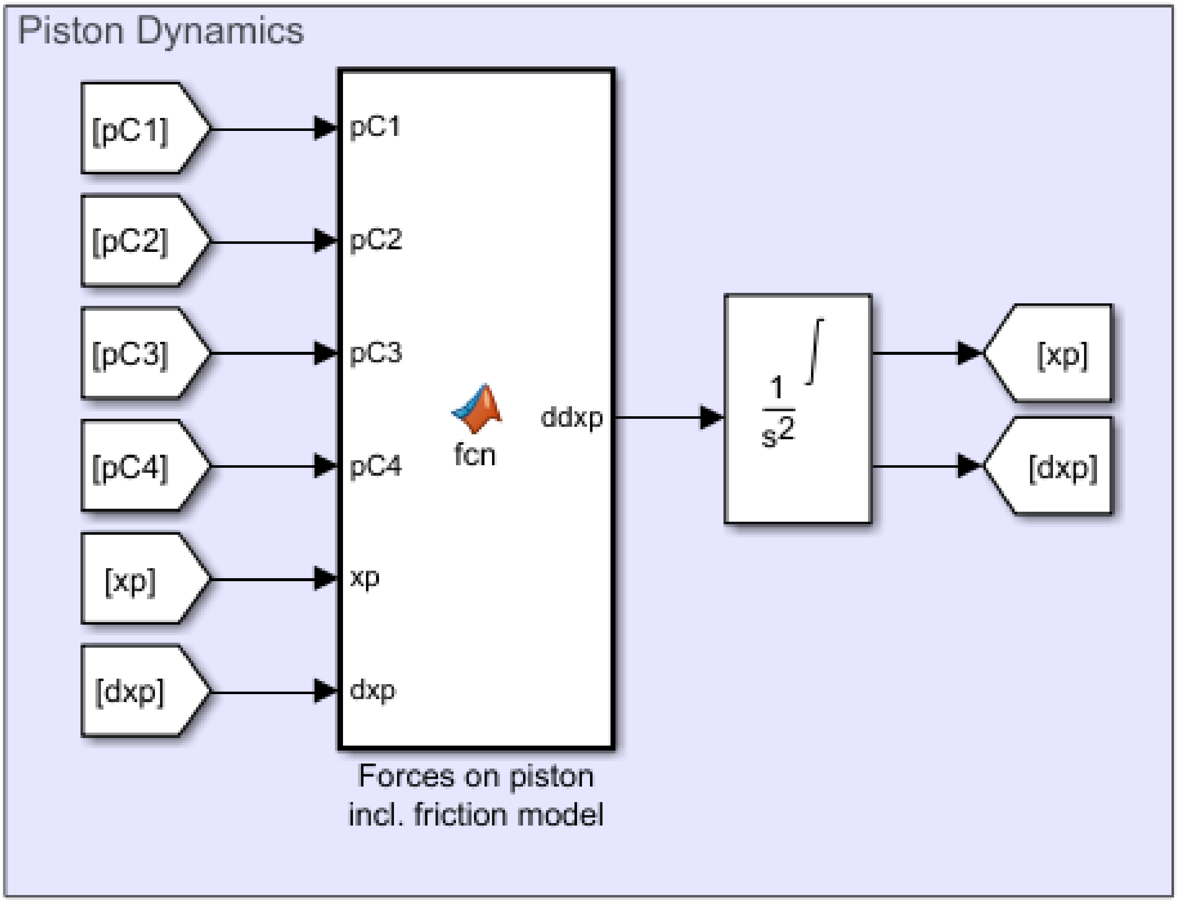


Figure B.8. Simulink Piston Dynamics

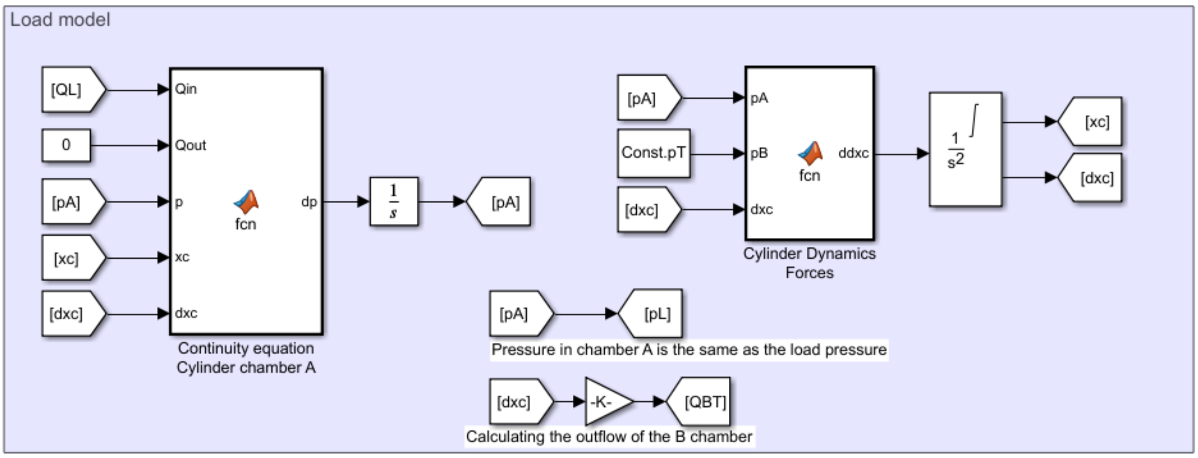


Figure B.9. Simulink Load Model

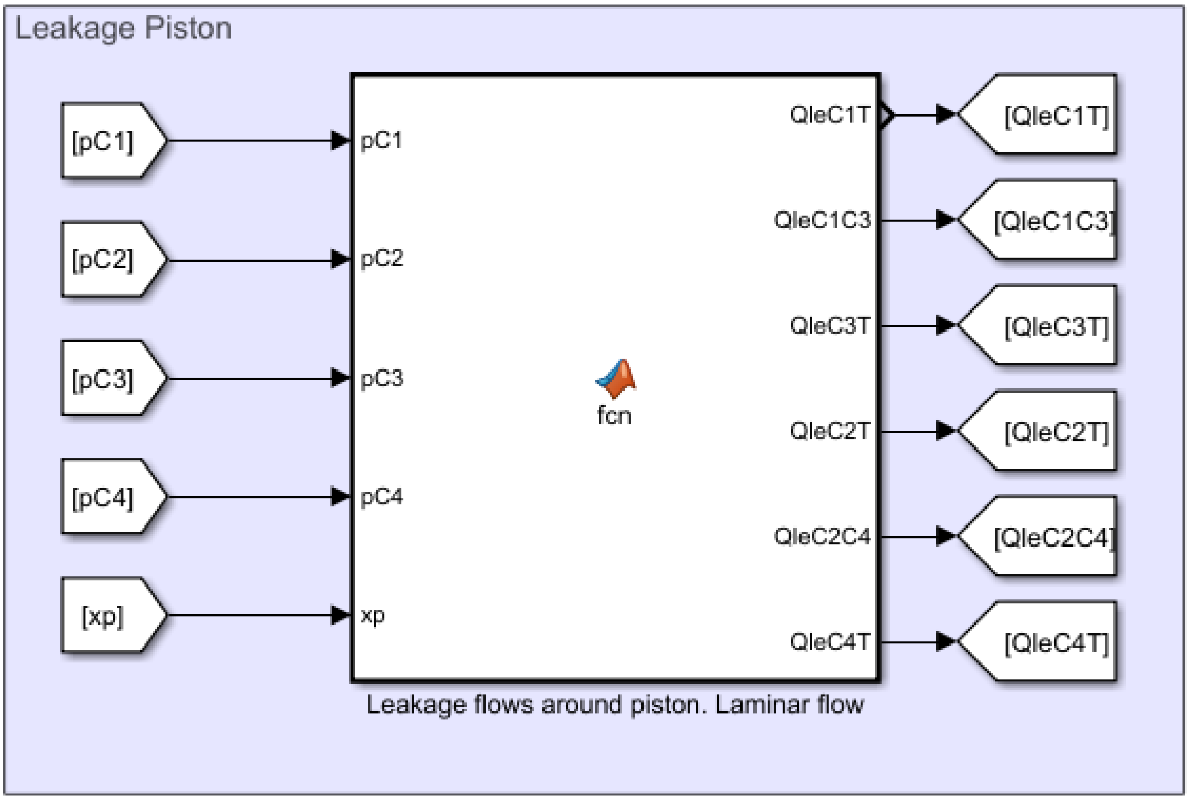


Figure B.10. Simulink Leakage Model FBOT chambers

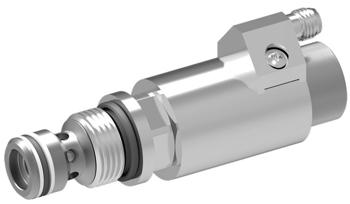
Appendix C
Data Sheets

C.1 On/Off Valve Data sheet

BUCHER
hydraulics

2/2 Cartridge Seat Valve, Size 5

Q_{max} = 30 l/min, p_{max} = 350 bar
Digital valve, bidirectional seat-valve shut-off, direct acting
Series WS22GD…/ WS22OD…

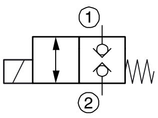


- For use in digital hydraulics
- With bidirectional seat-valve shut-off
- Compact construction for cavity type ALM – M20x1.5
- High switching performance
- Short response times
- All exposed parts with zinc-nickel plating
- High pressure wet-armature solenoids
- The slip-on coil can be rotated, and it can be replaced without opening the hydraulic envelope
- Can be fitted in a line-mounting body

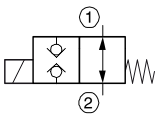
1 Description

These direct acting 2/2 solenoid operated directional seat valves, series WS22GD… / WS22OD…, are screw-in cartridges with a M20x1.5 or 3/4-16 UNF mounting thread. They are designed on the poppet/seat principle, and are therefore virtually leak-free in both directions of flow (bidirectional seat-valve shut-off). Over-excitation, preferably through an electronic switching device (booster), is required to operate the solenoid. Combined with the low mass of the moving parts, this results in short response times and high switching performance in a compact package. "De-energised closed" and "de-energised open" functions are available. The straightforward design delivers an outstanding price/performance ratio. The valves are used in applications in digital hydraulics, where fast response and long life with minimum size are vitally important. All external parts of the cartridge are zinc-nickel plated to DIN 50 979 and are thus suitable for use in the harshest operating environments. The slip-on coils can be replaced without opening the hydraulic envelope and can be positioned at any angle through 360°.

2 Symbol



WS22GD…



WS22OD…

3 Technical data

General characteristics	Description, value, unit
Designation	2/2 cartridge seat valve
Design	digital valve, bidirectional seat-valve shut-off, direct acting poppet/seat design (pressure balanced)
Mounting method	screw-in cartridge M20x1.5 or 3/4-16 UNF
Tightening torque	50 Nm ± 10 %
Size	nominal size 5, cavity type ALM M20x1.5 cavity type AL 3/4-16 UNF please contact BUCHER
Weight	0.20 kg

General characteristics	Description, value, unit
Mounting attitude	unrestricted
Ambient temperature range	-25°C…+80 °C

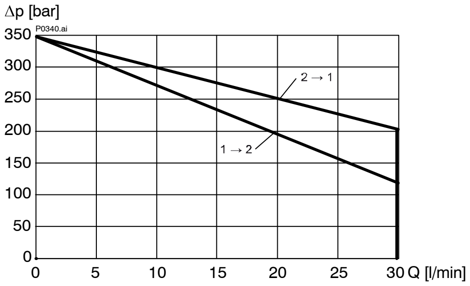
Hydraulic characteristics	Description, value, unit
Maximum operating pressure (ports 1 and 2)	350 bar
Maximum flow rate	30 l/min
Flow direction	1 → 2 / 2 → 1, see symbols
Hydraulic fluid	HL and HLP mineral oil to DIN 51 524; for other fluids, please contact BUCHER
Hydraulic fluid temperature range	-25 °C … +80 °C
Viscosity range	10…500 mm²/s (cSt), recommended 15...250 mm²/s (cSt)
Minimum fluid cleanliness Cleanliness class to ISO 4406 : 1999	class 20/18/15

Electrical characteristics	Description, value, unit
Excitation voltage	48 V DC (standard)
Length of over-excitation	4…5 ms
Supply voltage	12 V DC (standard)
Voltage tolerance	± 5 % (at ambient temperature < 60°C : ± 10 %)
Nominal power consumption	15 W at 12 V DC
Switching time - model WS22G…	6 … 20 ms (energising) 10 … 30 ms (deenergising)
- model WS22O…	6 … 30 ms (energising) 5 … 20 ms (deenergising) These times are strongly influenced by fluid pressure, flow rate and viscosity, as well as by the dwell time under pressure.
Relative duty cycle - static	100 %
Duty cycle / switching frequency - dynamic	see characteristics
Protection class to ISO 20 653 / EN 60 529	IP 65
Electrical connection:	3-pin plug M8x1
- PIN 1	48 / 12 V DC
- PIN 3	0 V
- PIN 4	not used

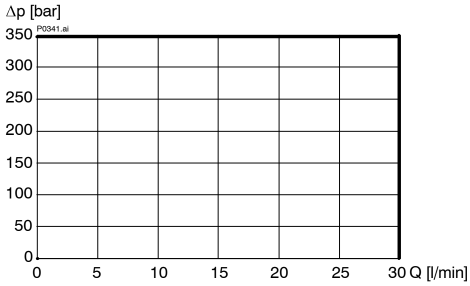
4 Performance graphs

measured with oil viscosity 33 mm²/s (cSt), coil at steady-state temperature and 10 % undervoltage

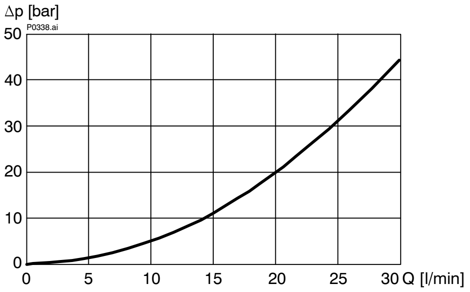
p = f (Q) Performance limits
[WS22GD…]



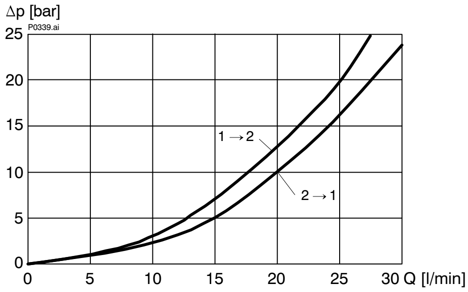
p = f (Q) Performance limits
[WS22OD…]



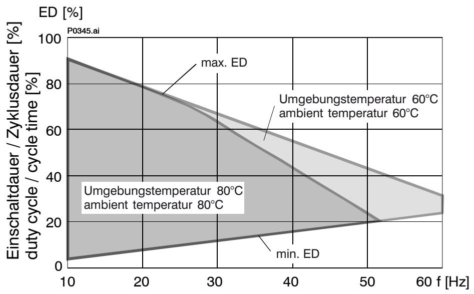
Δp = f (Q) Pressure drop - Flow rate characteristic
[WS22GD…]

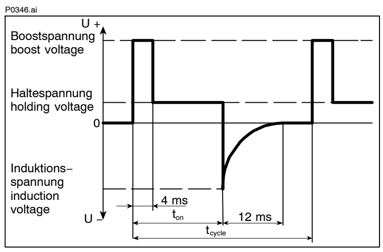


Δp = f (Q) Pressure drop - Flow rate characteristic
[WS22OD…]



ED = f (f) duty cycle - switching frequency - characteristic [at steady-state coil temperature]

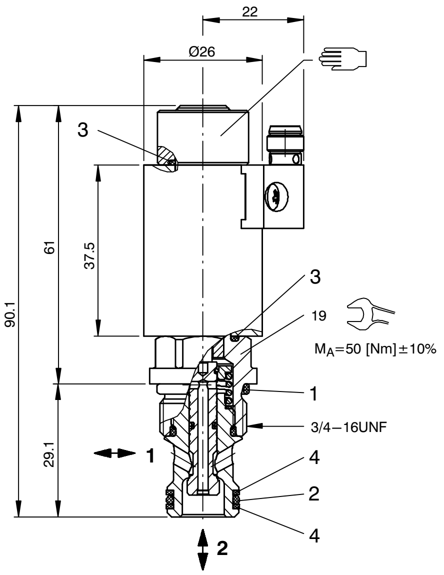




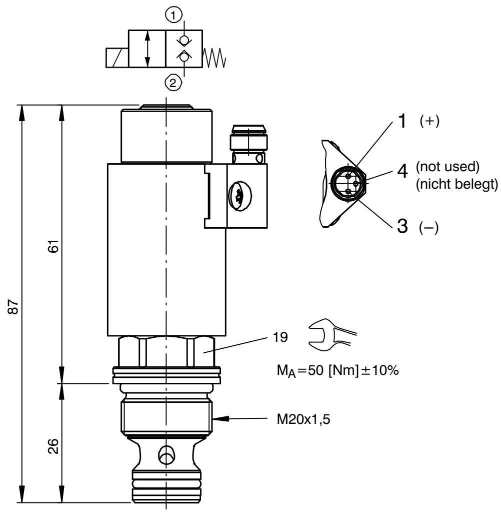
$$\text{ED [\%]} = \frac{t_{on}}{t_{cycle}} \times 100$$

5 Dimensions & sectional view

5.1 "Normally closed" design WS22GD…

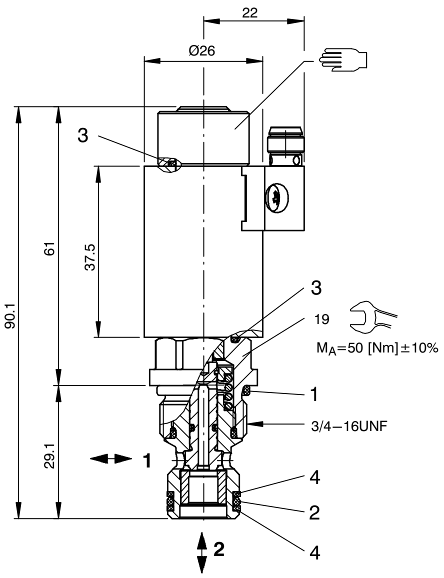


with 3/4-16 UNF thread – cavity type AL
please contact BUCHER

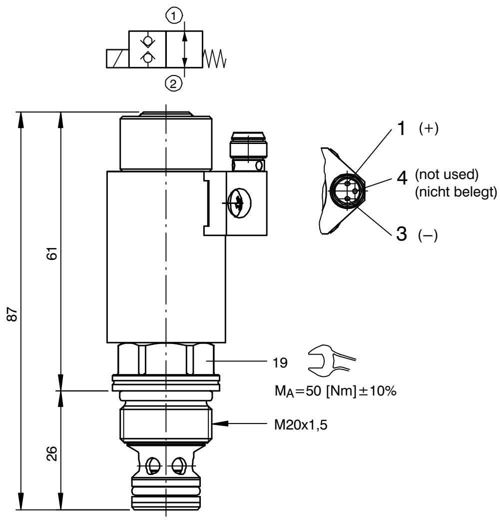


with M20x1.5 thread – cavity type ALM

5.2 "Normally open" design WS22OD…



with 3/4-16 UNF thread – cavity type AL
please contact BUCHER



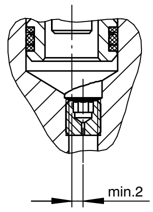
with M20x1.5 thread – cavity type ALM

6 Installation information

IMPORTANT!
When fitting the cartridges, use the specified tightening torque. No adjustments are necessary, since the cartridges are set in the factory.

ATTENTION!
Only qualified personnel with mechanical skills may carry out any maintenance work. Generally, the only work that should ever be undertaken is to check, and possibly replace, the seals. When changing seals, oil or grease the new seals thoroughly before fitting them.

ATTENTION!
If an orifice is fitted directly in port 2 close to the valve, and if the flow direction is from 2 to 1, it is important to ensure that the axis of the orifice drilling is offset from the valve axis by at least 2 mm!



3/4-16 UNF "A" – NBR seal kit no. DS-435-N [1)]

Item	Qty.	Description
1	1	O-ring no. 017 ∅ 17,17 x 1,78 N90
2	1	O-ring no. 014 ∅ 12,42 x 1,78 N90
3	2	O-ring ∅ 12.00 x 1.50 Viton
4	2	Backup ring ∅10.70 x 1,45 x 1,0 FI0751

M20x1.5 "Z" - NBR seal kit no. DS-436-N [1)]

Item	Qty.	Description
1	1	O-ring no. 017 ∅ 17,17 x 1,78 N90
2	1	O-ring no. 013 ∅ 10,82 x 1,78 N90
3	2	O-ring ∅ 12.00 x 1.50 Viton
4	2	Backup ring ∅ 9.90 x 1,45 x 1,4 FI0751

IMPORTANT!
1) Seal kit with FKM (Viton) seals, no. DS-435-V

IMPORTANT!
1) Seal kit with FKM (Viton) seals, no. DS-436-V

7 Ordering code

Ex. | W | S | 22G | D | Z | 5 | _ | - | 1 | 12 | D | _ |

W = directional valve
S = seat valve, direct acting
22G = 2/2 function, de-energised closed
22O = 2/2 function, de-energised open
D = digital valve
Z = special features - with M20x1.5 thread (**standard**)
A = standard model - with 3/4 - 16 UNF thread (please contact Bucher)
5 = nominal size 5
(blank) = NBR (Nitrile) seals (**standard**)
V = FKM (Viton) seals
(special seals - please contact BUCHER)
1 ... 9 = design stage (omit when ordering new units)
... = voltage e.g. 12 (12 V)
D = current DC
(blank) = M8x1 male connector (**standard**)
F = for flying leads (1000 mm), please contact Bucher

BUCHER
hydraulics

8 Related data sheets

Reference	(Old no.)	Description
400-P-040011	(i-32)	The form-tool hire programme
400-P-040171	(i-33.10)	Cavity type AL
400-P-040201	(i-33.13)	Cavity type ALM
400-P-720101	(G-4.10)	Line-mounting body, type GALA (G 3/8")
400-P-720105	(G-4.11)	Line-mounting body, type GALMA (M20 x 1.5)

info.ch@bucherhydraulics.com www.bucherhydraulics.com

Classification: 430.300.-.305.305.300

Appendix D
Multi-objective Cost function

D.1 Objective Function

The objective function is also called the cost function, and optimization aims to minimize this function. For this control scheme, the cost function is multi objective although it would be more effective to only have a single cost. The cost consists of 1. positional cost, 2. cost of not using the output pressure to absorb kinetic energy and 3. not turning on HP braking pressure before the positional peak is reached. Each cost is elaborated, and the total cost term is

$$J(\boldsymbol{t}_{des}) = J_1(\boldsymbol{t}_{des}) + J_2(\boldsymbol{t}_{des}) + J_3(\boldsymbol{t}_{des}) \tag{D.1}$$

Note that all cost terms implicitly are functions of the design variables, but the notation is dropped through the rest of the cost description.

J_1 **Positional vs. Reference**
Inside the cost function, a prediction is run. The piston trajectory is evaluated for the maximum value if the reference is high and the minimum value if the reference is low. The time at which the peak occurs is stored as k_{peak}, and piston position at peak is $x_{p,k_{peak}}$. The cost can then be defined as a quadratic expression:

$$J_1 = W_{1,t}(x_{p,k_{peak}} - x_{ref})^2 \tag{D.2}$$

The quadratic term is necessary such that the error becomes positive in all cases. An alternative formulation is to compare the whole trajectory with the reference, which is also a viable strategy. Peak value evaluation is can be more effective since the optimization scheme should not consider what happens after the peak has been reached within a single prediction horizon. On the other hand, it can result in a slower trajectory, since there is no cost enticement as to what speed or time the reference is reached which would be a natural part of evaluating the whole trajectory. The weighting term $W_{1,t}$ is comprised of a normalizing term and a tuning weight parameter as

$$W_{1,t} = W_1 \frac{1}{(0.025)^2} \tag{D.3}$$

The fractional term which is normalizing is based on the maximum error that could be experienced if the piston stays within the *high* and *low* positions. W_1 is a tuning parameter and is initially a high value of 1000. Impacts of tuning are explored later.

J_2 **Output Pressure Absorption**
There must be some enticement for the optimizer to use the output pressure as much as possible to brake the piston, such that as much kinetic energy as possible is transformed to the output manifold. The optimizer could choose to primarily use the braking pressure, but this should be used to the smallest degree possible.

The first step in evaluating this cost is to identify which pressures acts as the different pressure roles based on operating mode. This is chosen from table 5.2.

The next step is to identify when the output pressure becomes high i.e. near manifold pressure level, and when it becomes low again. This results in two time stamps respectively; $k_{O,HP,1}$ and $k_{O,HP,end}$. The effective output time is the difference of these two time stamps; $T_{O,HP} = T_s(k_{O,HP,end} - k_{O,HP,1})$. This term should be as large as possible, so the cost is defined as its inverse. It is however important to make sure that there are no divisions by zero, which is handled by checking whether the time is zero, and if it is, then the time is replaced by a small number. The cost is then defined as:

$$J_2 = W_2 \frac{1}{T_s} \frac{1}{T_{O,HP}} \tag{D.4}$$

where W_2 is a weighting function set initially to 100. The inverse of the sampling time is used as a normalization term.

J_3 **Braking Pressure Timing**

The third cost term regards the braking pressure. This pressure should be high i.e. near manifold pressure, when the positional peak is reached. Table 5.2 is used to determine the braking pressure and its manifold value, and similarly with cost 2, the time stamps of high pressure are evaluated, such that $T_{B,HP,on}$ is the time when the braking pressure becomes high. The cost is then the difference between when the pressure becomes high and the peak as:

$$J_3 = W_3 \frac{1}{T_s} (T_{B,HP,on} - T_s k_{peak})^2 \tag{D.5}$$

which is a quadratic expression such that the difference is always positive. W_3 is again a weighting term which initially is set to 1.

D.2 Tuning of Cost Function Weights

The MPC is built up around an optimization scheme which includes weighting in the cost function. This subsection will dwell on the impact of tuning weights. The tuneable weights are: Position to reference $J1W$ and a cost on using the braking pressure $J2W$.

The cost terms have been normalized to a degree but cannot be directly compared, so the influence of either increasing or decreasing weight is considered rather than the meaning of specific tuning values.

Tuning of J_1 has a direct impact on how well the reference position is followed which is seen in the tuning figure D.1. The positional cost weighting is varied from 1 to 100000 while the braking pressure cost is at 1. It can be seen that when J_1 is increased, the peak of x_p comes closer to the reference. For small values, the positional constraints are violated, and for the two largest values, the position never exceeds the reference. The two largest weights show little difference in regards to position, and when the weights are so large, the other cost term is practically nullified and the cost function can approximately be seen as a single objective cost function. A weight of 1000 gives an adequate positional response and will be held constant while varying the braking cost term J_2, which is shown in figure D.2. This cost term should primarily have an influence on pressure timings, so the graph includes three chamber pressures p_1, p_4 and p_2 which respectively has the roles of input, output and braking. The first weight is zero which practically corresponds to the largest weighting of the previous tuning case in the sense that the cost function boils down to being single objective. This shows that the braking cost term is not strictly necessary to have a functioning FBoT, but that makes sense, since the cost mainly exists to increase the efficiency of the FBoT. Tuning of J_2 has little influence at low values, and for higher values, the FBoT performs worse. At higher weightings, the output pressure is diminished and the position violates constraints and the input pressure is active for too long time. This means that the term is unnecessary and adverse. The MPC will be better with a single objective cost function.

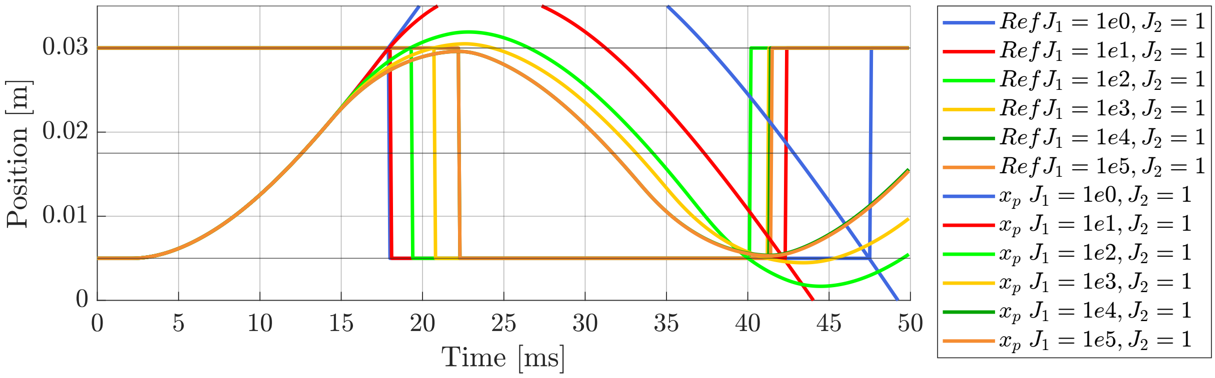


Figure D.1. Tuning of J_1 Positional cost weighting

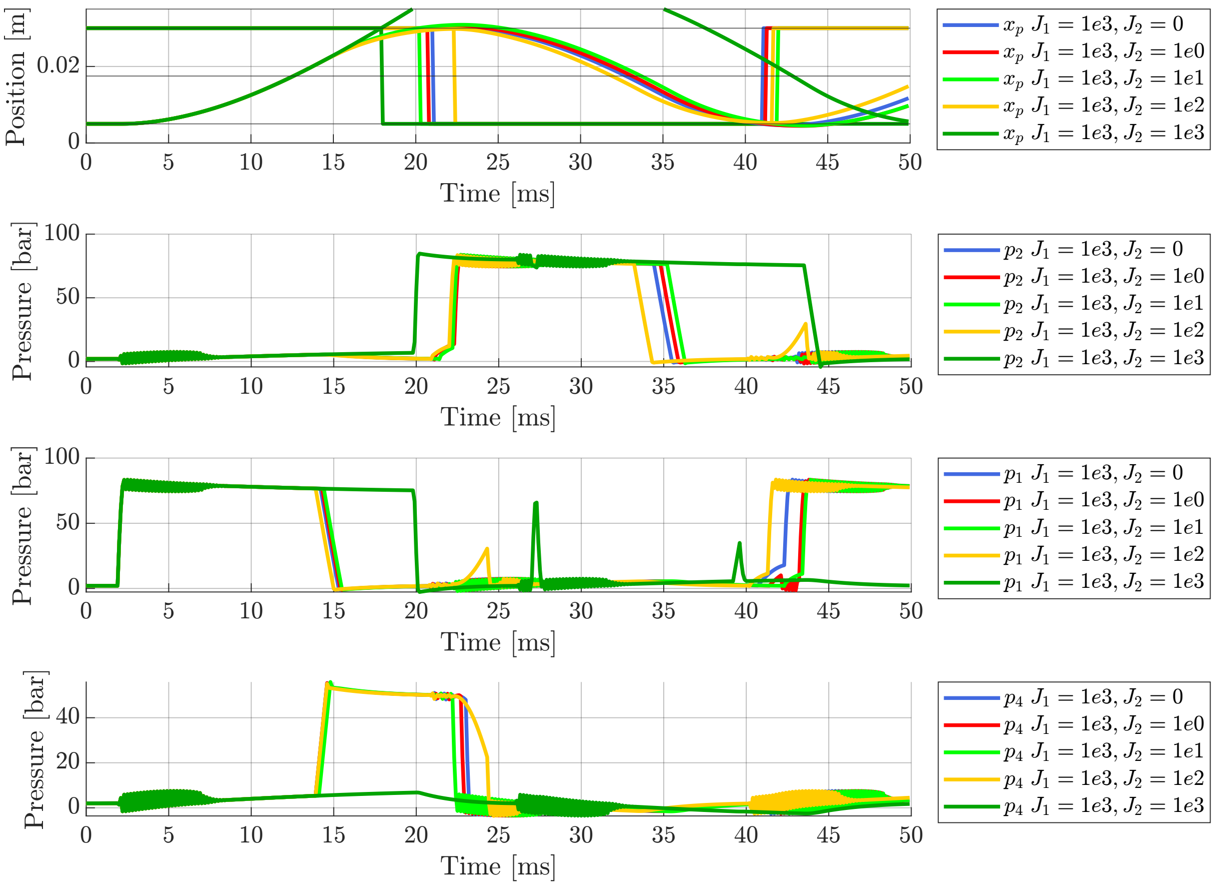


Figure D.2. Tuning of J_2 Braking pressure cost weighting

Appendix E
MATLAB SCRIPTS

E.1 Main

```
%%% MPC development %%%
% Main Document which runs a plant simulation with MPC
clear; clc; close all;

Ts  = 0.1*10^-3; % [s] Step time for control system
Tss = 0.1*10^-3; % Simulation step time.

nx = 6;          % Number of states: x, dx, p1, p2, p3, p4
nu = 8;          % Number of inputs: u1 to u8
np = 100;        % Prediction Horizon in time steps
nt = 1000;       % Simulation time in time steps

k_now   = 1;       % Current simulation time step
t = 0:Ts:(nt)*Ts; % Simulation time

% Constant parameters - but manifold pressures should be measured!
% Everything SI units
low     = 5*10^-3;       % Position lower bound
high    = 30*10^-3;      % Position upper bound
p0      = 2*10^5;        % Tank pressure
pS      = 80*10^5;       % Supply pressure
pL      = 50*10^5;       % Load pressure
t_lag   = 2*10^-3;       % [s] valve lag time for simulating delay dynamics
h       = 25*10^-6;      % Clearence between piston and house, (sliding)
mu0     = 0.4002;        % Viscosity
As      = 0.0067;        % Sliding area between piston and chamber
B       = As*mu0/h;      % Viscous Damping term
A_set   = (1.22e-04)/2; % Piston pressure area

q0      = [5*10^-3 0.0001 p0 p0 p0 p0];      % System initial state
                                             % [pos vel p1 p2 p3 p4]
% 0 = regenerate
Pump    = 1; % Choosing operating mode. Pumping mode or regenerating mode.
xref    = high; % Initial target reference
trigger = [0 0];% Initialize trigger

% Parameter vector which is used frequently inside functions
params = [Ts np k_now xref p0 pS pL B t_lag Pump Tss trigger];

% Initialising global variables which are used in optimization
global nDesVar Pert nIter

nDesVar = 2;         % Design variables aka. switching times - which we want to optimize!
Pert    = 1*10^-3;   % Perturbation of switching times
nIter   = 0;         % Number of iterations initialization

%% Constraints: Bounds and Linear Constraints
```

```
lb(1:2,1) = Ts;          % Lower bound on switching times. Smallest time is a single time sample. If zero then error
ub(1:2,1) = Ts*(np-1);  % Upper bound is dictated by the prediciton horizon
ub(2,1) = Ts*(np-1) - t_lag; % Brake HP turn on cannot turn on too late.
% Linear inequality constraints
    Acon    = [1 -1 ];  bcon    = [0];

% Options for optimization
options = optimoptions('fmincon', 'Algorithm','sqp', 'MaxFunEvals',1000, ...
                       'SpecifyObjectiveGradient',true,'StepTolerance',1e-5, 'Display', 'off','SpecifyConstraintGradient',true);
%% Simulation Initialization
% State and input history matrices are initialized
Qhistory    = zeros(nt,nx);
Uhistory    = zeros(nt,nu);
Zhistory    = zeros(nt,nu);
Refhistory  = zeros(nt,1);
WIhistory   = zeros(nt,1);
WOhistory   = zeros(nt,1);

% To simulate valve delay dynamics, and ideal delay of inputs are created.
lag         = round(params(9)/Ts);  % time samples delay. Round to ensure integer

u_lagged = zeros(lag,8);
U_past = zeros(lag,8);
dq_old = 0;
WI = 0;
WO = 0;

tic % Start clock
for i = 1:nt % Simulate for nt time steps
    fprintf('Simulation step: '),disp(i);
    nIter = 0;               % Number of iterations reset before every optimization

    %% Choosing design variables depending on whether pump/regen mode AND whether target up/down
    if q0(1)    <= 5*10^-3 || (sign(dq_old) ~=  sign(q0(2)) && q0(1)    <= 17.5*10^-3)% If at/below threshold, then target is opposite direction
        xref    = high; % setting target position
        target  = 1;    % setting target direction
        trigger = [0 0] % 0: Initialize, 1: detect turn on, 2: detect turn off. pI and pB respectively
        if Pump == 1    % in pump mode
            xdes_idx    = [2 3]'; % [shoot_off brake on] index of desvar in xdes_pool
            xdes        = 10^-3 * [ [7 11] ]'; % Initial guess
        else
            xdes_idx    = [6 7]';
            xdes        = 10^-3 * [ [ 7 17] ]';
        end
    end
    if  q0(1)   >= 30*10^-3 || (sign(dq_old) ~=  sign(q0(2)) && q0(1)    >= 17.5*10^-3)
        target  = -1;
        xref    = low;
        trigger = [0 0]
        if Pump == 1
```

```
            xdes_idx    = [4 1]';
            xdes        = 10^-3 * [ [7 17] ]';
        else
            xdes_idx    = [8 5]';
            xdes        = 10^-3 * [ [7 17] ]';
        end
    end

    dq_old = q0(2); % Old value is stored for next iteration
    %Updating Parameters
    params(4)   = xref;
    params(3)   = i; % Current time step
    params(12:13) = trigger;

    % Constraint and objective function is initialized with passed parameters
    Constraints = @(xdesvar)  IneqConstraints(xdesvar,q0,params,U_past);
    f_opt       = @(xdesvar)  CallFunctionObjAndGrad(xdesvar,q0,params,xdes_idx,U_past);

    % Optimization of problem
    [xdes,fval,exitflag,output,LAMBDA,GRAD,HESSIAN] = fmincon(f_opt,xdes,Acon,bcon,[],[],lb,ub,Constraints,options);

    xdes_pool   = xdes2pool(xdes,params); % Switching times are generated
    U           = tswitch2inputs(xdes_pool,params);% Function generates matrix of input signals from switching times

    q0_debug = q0; % Saving this before progressing. Just for debugging simulation
    q_old = q0;
    u = U(1,:); % Input vector from first row of input matrx
    Z = [U_past ; U]; %Constructing delayed valve signals
    z = floor(mean([Z(1,:); Z(2,:)])); % Smoothing the delayed signal to avoid small spikes

    [qdot]  = SystemPLANT(q0,z,params);     % Inputs are fed into the plant, and differential equations are evaluated
    q       = q0 + Tss*qdot;                % States are updated - forwards euler integration
    q0      = q;                            % New initial state for next simulation step time

    U_past = circshift(U_past,-1);          % Shifting delayed input
    U_past(lag,:) = u;

    % Identifying the input functions :))
    [uI_id, uO_id, uB_id, pI_idx, pO_idx, pI_HP, pO_HP] = idFunc(Pump,target,pS,pL);
    % Detecting trigger
    trigger = SwitchTrigger(U,uI_id,uB_id, trigger);

    % Evaluate work.
    [WO, WI] = Work(A_set,q0,q_old,WO,WI);
    WIhistory(i) = WI;
    WOhistory(i) = WO;

    % Results are stored in history matrix
    Qhistory(i,:)    = q0;               % States
    Uhistory(i,:)    = u;                % Inputs
    Zhistory(i,:)    = z;                % Lagged input
```

```
    Refhistory(i)    = xref;             % Logging current reference

end
toc

%% Debug Plot

xdes_pool    = xdes2pool(xdes,params);
[Q,U,Z]      = Simulator(q0_debug,xdes_pool,params,U_past);
Debug_Figure(Q,U(1:np,:),t,np,target,Pump,low,high,xref)

%% Plots
figure(1)
tiles = tiledlayout(8,1,'TileSpacing','tight','Padding','tight');


nexttile([2 1])
hold on
plot(t(1:nt)*1000,Refhistory(1:nt))
plot(t(1:nt)*1000,Qhistory(:,1))
ylim([0 35*10^-3])
xlim([0 t(end)*1000])
yline(low)
yline(high)
yline(35*10^-3)
yline(17.5*10^-3)
legend('Ref','x sim','','','','Location','east')
xlabel('Time [ms]')
ylabel('Position [m]')
xticks(0:5:nt/Ts);

nexttile([2 1])
hold on
plot(t(1:nt)*1000,Qhistory(:,3:6)*10^-5)
xlim([0 t(end)*1000])
legend('p1','p2','p3','p4','Location','east')
xlabel('Time [ms]')
ylabel('Pressures [bar]')
xticks(0:5:nt/Ts);

nexttile
hold on
plot(t(1:nt)*1000,Uhistory(:,1),'color',[65, 105, 225]/255)
plot(t(1:nt)*1000,Uhistory(:,5),'--','color',[65, 105, 225]/255)
plot(t(1:nt)*1000,Zhistory(:,1)+2,'color','black','Linewidth',1)
plot(t(1:nt)*1000,Zhistory(:,5)+2,'--','color','black','Linewidth',1)
legend('u1','u5','z1','z5','Location','east')
xlim([0 t(end)*1000])
xticks(0:5:nt/Ts);
yticklabels([0 1 0 1])
grid on

nexttile
hold on
plot(t(1:nt)*1000,Uhistory(:,2),'color',[255, 0, 0]/255)
plot(t(1:nt)*1000,Uhistory(:,6),'--','color',[255, 0, 0]/255)
plot(t(1:nt)*1000,Zhistory(:,2)+2,'color','black','Linewidth',1)
plot(t(1:nt)*1000,Zhistory(:,6)+2,'--','color','black','Linewidth',1)
```

```
legend('u2','u6','z2','z6','Location','east')
xlim([0 t(end)*1000])
xticks(0:5:nt/Ts);
yticklabels([0 1 0 1])
grid on

nexttile
hold on
plot(t(1:nt)*1000,Uhistory(:,3),'color',[0, 255, 0]/255)
plot(t(1:nt)*1000,Uhistory(:,7),'--','color',[0, 255, 0]/255)
plot(t(1:nt)*1000,Zhistory(:,3)+2,'color','black','Linewidth',1)
plot(t(1:nt)*1000,Zhistory(:,7)+2,'--','color','black','Linewidth',1)
legend('u3','u7','z3','z7','Location','east')
xlim([0 t(end)*1000])
xticks(0:5:nt/Ts);
yticklabels([0 1 0 1])
grid on

nexttile
hold on
plot(t(1:nt)*1000,Uhistory(:,4),'color',[255, 204, 0]/255)
plot(t(1:nt)*1000,Uhistory(:,8),'--','color',[255, 204, 0]/255)
plot(t(1:nt)*1000,Zhistory(:,4)+2,'color','black','Linewidth',1)
plot(t(1:nt)*1000,Zhistory(:,8)+2,'--','color','black','Linewidth',1)
legend('u4','u8','z4','z8','Location','east')
xlim([0 t(end)*1000])
xticks(0:5:nt/Ts);
yticklabels([0 1 0 1])
grid on
```

E.2 Objective Function Gradient

```
    function [f, c] = CallFunctionObjAndGrad(xdes,q0,params,xdes_idx,U_past)
% This function evalutes objective function and gradients

global nDesVar Pert nIter
Ts = params(1);

f = CostFun(xdes,q0,params,U_past);   % Cost function is evaluated

if (nargout > 1)
    nIter = nIter + 1;
        for DVNo = 1:nDesVar
            if ~rem(DVNo, 2) == 0
                xdes(DVNo) = xdes(DVNo) + Pert;                   % Perturb the
                    design variable by Pert
                c(DVNo) = (CostFun(xdes,q0,params,U_past) - f) / Pert; % Approximate
                     the gradient c by forward difference
                xdes(DVNo) = xdes(DVNo) - Pert;                  % Restore the
                    original value
            else
                xdes(DVNo) = xdes(DVNo) - Pert;                   % Perturb the
                    design variable by Pert
                c(DVNo) = -(CostFun(xdes,q0,params,U_past) - f) / Pert; %
                    Approximate the gradient c by forward difference
```

```
                xdes(DVNo) = xdes(DVNo) + Pert;                 % Restore the
                    original value
            end
        end
end

end
```

E.3 Case Identifier

```
    function [xIHP, xILP, xOLP, xBHP, xBLP] = Cases(Pump,target)

if target == 1 %
    if Pump ==1
        xIHP  = [1  2 ]; % [on off]
        xILP = [9  10];  % [off on]
        xOLP = [15 16];  % [off on]
        xBHP  = [3  4 ]; % [on off]
        xBLP = [11 12];  % [off on]
    else % Regen
        xIHP  = [5  6 ]; % [on off]
        xILP = [13 14];  % [off on]
        xOLP = [11 12];  % [off on]
        xBHP  = [7  8 ]; % [on off]
        xBLP = [15 16];  % [off on]
    end
else % Target -1
    if Pump ==1
        xIHP  = [3  4 ]; % [on off]
        xILP = [11 12];  % [off on]
        xOLP = [13 14];  % [off on]
        xBHP  = [1  2 ]; % [on off]
        xBLP = [9  10];  % [off on]
    else % Regen
        xIHP  = [7  8 ]; % [on off]
        xILP = [15 16];  % [off on]
        xOLP = [9  10];  % [off on]
        xBHP  = [5  6 ]; % [on off]
        xBLP = [13 14];  % [off on]
    end
end
```

E.4 Cost Function

```
    function J = CostFun(xdes,q0,params,U_past)

xdes_pool = xdes2pool(xdes,params);
[Q,~,~] = Simulator(q0,xdes_pool,params,U_past); % States are simulated for
    prediction horizon

% Parameters are loaded
xref = params(4);
target = xref > 17.5*10^-3;
```

```
% Identifying peak
if target == 1
    k_peak = find(max(Q(:,1)) == Q(:,1));
else
    k_peak = find(min(Q(:,1)) == Q(:,1));
end
% We don't need to include what happens after the peak in the cost.
%% Weighting, Normalization and Evaluation of costs, incl. pos, vel costs.

% Position, (compared to reference during prediction horizon until peak)
J1_W    = 1000;                                         % Weight
J1_m    = 1/((25*10^-3)^2);                             % Normalization term. Maximum error possible if within constraints
J1      = (J1_W*J1_m*(((Q(k_peak,1)) - xref).^2));   % quadratic error from ref

J = J1 ;
```

E.5 Debug Figure

```
    function [] = Debug_Figure(Q,U,t,np,target,Pump,low,high,xref)

if target == 1
    if Pump == 1
        u_I = 1;
        u_O = 8;
        u_B  = 2;
    else
        u_I = 3;
        u_O = 6;
        u_B  = 4;
    end
else
    if Pump == 1
        u_I = 2;
        u_O = 7;
        u_B  = 1;
    else
        u_I = 4;
        u_O = 7;
        u_B  = 3;
    end
end

figure(2)

subplot(4,2,1)
hold on
plot(t(1:np)*1000,Q(:,1))
ylim([0 35*10^-3])
xlim([0 t(np)*1000])
yline(low)
yline(high)
yline(35*10^-3)
yline(17.5*10^-3)
```

```
yline(xref,'r')
legend('x sim','','','','','xref')
xlabel('Time [ms]')
ylabel('Position [m]')

subplot(4,2,3)
hold on
plot(t(1:np)*1000,U(:,u_I),'color','blue')
plot(t(1:np)*1000,U(:,u_O),'color','magenta')
plot(t(1:np)*1000,U(:,u_B),'color','green')
legend('u1','u5','u8')
xlim([0 t(np)*1000])
grid on

subplot(4,2,5)
% hold on
% plot(t(1:np)*1000,U(:,2),'color','red')
% plot(t(1:np)*1000,U(:,6),'color','yellow')
% plot(t(1:np)*1000,U(:,7),'color','green')
% legend('u2','u6','u7')

subplot(4,2,7)
hold on
plot(t(1:np)*1000,Q(:,3:6)*10^-5)
xlim([0 t(np)*1000])
legend('p1','p2','p3','p4')
xlabel('Time [ms]')
ylabel('Pressures [bar]')

subplot(4,2,2)
hold on
plot(t(1:np)*1000,U(:,1),'color','blue')
plot(t(1:np)*1000,U(:,5),'--','color','blue')
legend('u1','u5')
xlim([0 t(np)*1000])
grid on

subplot(4,2,4)
hold on
plot(t(1:np)*1000,U(:,2),'color','red')
plot(t(1:np)*1000,U(:,6),'--','color','red')
legend('u2','u6')
xlim([0 t(np)*1000])
grid on

subplot(4,2,6)
hold on
plot(t(1:np)*1000,U(:,3),'color','green')
plot(t(1:np)*1000,U(:,7),'--','color','green')
legend('u3','u7')
xlim([0 t(np)*1000])
grid on

subplot(4,2,8)
hold on
plot(t(1:np)*1000,U(:,4),'color','yellow')
plot(t(1:np)*1000,U(:,8),'--','color','yellow')
legend('u4','u8')
```

```
xlim([0 t(np)*1000])
grid on
```

E.6 Role Identifier

```
    function [uI_id, uO_id, uB_id, pI_idx, pO_idx, pI_HP, pO_HP] = idFunc(Pump,
        target,pS,pL)

if target == 1
    if Pump == 1
        uI_id = 1;
        uO_id = 8;
        uB_id = [2 6];
        pI_idx = 3; % in q
        pO_idx = 6;
        pI_HP = pS;
        pO_HP = pL;
    else
        uI_id = 3;
        uO_id = 6;
        uB_id = [4 8];
        pI_idx = 5; % in q
        pO_idx = 4;
        pI_HP = pL;
        pO_HP = pS;
    end
else
    if Pump == 1
        uI_id = 2;
        uO_id = 7;
        uB_id = [1 5];
        pI_idx = 4; % in q
        pO_idx = 5;
        pI_HP = pS;
        pO_HP = pL;
    else
        uI_id = 4;
        uO_id = 5;
        uB_id = [3 7];
        pI_idx = 6; % in q
        pO_idx = 3;
        pI_HP = pL;
        pO_HP = pS;
    end
end
```

E.7 Inequality Constraints

```
    function [cineq, ceq, grad,DC_eq] = IneqConstraints(xdes,q0,params,U_past)
DC_eq = [];
xdes_pool = xdes2pool(xdes,params);

% Constraints such that
```

```
% 1. Position of piston is within bounds
% 2. Inputs are only turned on during either high or low pressure. This is
% only really relevant for the braking pressure since the input pressure
% just should turn on instantly.

% cineq: inequality, ceq: equality, DC: inequality gradient, DCeq:
% equality gradient

% Simulate states and inputs from current position and design variables for
% prediction horizon
[Q,~,~] = Simulator(q0,xdes_pool,params,U_past);

% Parameters are loaded
p       = params(2);
p_tol   = 10*10^5;    % [Pa] pressure tolerance
Pump    = params(10);
pS      = params(6);
pL      = params(7);
xref    = params(4);
target  = xref > 17.5*10^-3;

% Identifying peak
if target == 1
    k_peak = find(max(Q(:,1)) == Q(:,1));
else
    k_peak = find(min(Q(:,1)) == Q(:,1));
end
% We get the current pressures from chambers 1 and 2,3,4
pc1 = Q(:,3); pc2 = Q(:,4); pc3 = Q(:,5); pc4 = Q(:,6);

if Pump == 1
    if target == 1; pB = pc2; pBH = pS;
    else;           pB = pc1; pBH = pS;
    end
else % Regeneration
    if target == 1; pB = pc4; pBH = pL;
    else;           pB = pc3; pBH = pL;
    end
end

%% Constraint 1
% Create vector of positions exceeding bounds.
% The values are summed.
Qpos1 = abs(Q(Q(1:k_peak,1)-34*10^-3 > 0));
Qpos2 = abs(Q(1*10^-3-Q(1:k_peak,1) > 0));
cQ = sum([Qpos1;Qpos2]);

%% Constraint 2 Valve opening timing in regards to Pressure
% If a "turn on" time is NOT part of the manifold
% pressure time signatures. Further from good time = worse constraint

% Find out when the pressure is high
idx_pBHP = find( (pBH - p_tol <= pB  ) );

if isempty(idx_pBHP);     idx_pBHP = p; end % If it is empty, it is never high, so
    it's too late

% Is the first time instant of high pressure the same time as the peak?
```

```
c_type1 =  ~((idx_pBHP(1) == (k_peak))); % Evaluate whether constraint is violated.

% We figure out how far off the turn on point is.
% But let's do it smart: The pressure should be high at peak. Then it is okay for the next-cycle input to turn on instantly.
% That is the constraint
grad_c = 0;
if c_type1 ~= 0
    if idx_pBHP(1) - (k_peak)  < 0 % Turns on too early. Should turn on later
        c_type1 =  idx_pBHP(1) - (k_peak);
        grad_c = c_type1; % Estimate a gradient.
    elseif  (k_peak) - idx_pBHP(end) < 0 % Turns on too late. Should turn on sooner
        c_type1 =  (k_peak) - idx_pBHP(1);
        grad_c = -c_type1;
    end
    % It is probable turning on too late, so we find time difference
    % between first high pressure point and the turn on time
    %c_type1 =  idx_pBHP(1) - (kxdes  + k_lag)
end

% We multiply with an appendix, such that if the design values are on the
% extreme values, then the constraint is disregarded.
% Constraint finished
W_grad = 0.001;
W_cons = 1;

grad = [0 0;
        0 W_grad*grad_c];

        cineq = [   cQ;
                    abs(W_cons*0.01*c_type1)];
        ceq = [];
```

E.8 Simulator

```
    function [Q,U,Z_hist] = Simulator(q0,xdes_pool,params,U_past)

%% Loading parameters and initializing matrices, generating input matrix
    Tss = params(11); % Simulation step time
    p   = params(2);
    nx  = length(q0);
    U   = tswitch2inputs(xdes_pool,params); % Generating Input matrix for whole simulation time
                                            % based on switching times.
    q = q0;
    Q = zeros(p,nx);
    Z_hist = zeros(p,8);                     % Lagged inputs are saved for export
%%  Simulating valve dynamics as ideal lag
   Z = [U_past ; U];

    for i = 1:p                % Simulating for prediction horizon
        Q(i,:) = q;
        z = Z(i,:);            % The lagged input is fed into the system
        for j = 1:1
            q = q + Tss*System(q,z,params);
```

```
        end
        Z_hist(i,1:8) = z;
    end
end % end function
```

E.9 Switch Trigger

```
    function trigger = SwitchTrigger(U,uI_id,uB_id, trigger)

% Monitoring input value
    uI = U(1,uI_id);
    uB = U(1,uB_id);
    if uI == 1
        if trigger(1) == 0
            trigger(1) = 1;
        end
    end
    if uB(1) == 1
        if trigger(2) == 0
            trigger(2) = 1;
        end
    end
    if uI == 0
        if trigger(1) == 1
            trigger(1) = 2;
        end
    end
    if uB(1) == 0
        if trigger(2) == 1
            trigger(2) = 2;
        end
    end
```

E.10 Prediciton Model

```
    function qdot = System(q,u,params)
%%% System_reduced %%%

%% Getting constants - parameters

p0 = params(5);              % Tank pressure
pS = params(6);              % Supply pressure
pL = params(7);              % Load pressure

% System constants are loaded
A = (1.22e-04)/2;            % Piston area
m = 1101.79e-3+417.41e-3;    % mass piston
B = params(8);               % Damping term
beta = 1.4000e+09;           % Bulk modulus. We could have term for evaluating bulk
    modulus for current pressures.
V = 2.6996e-05;              % Volume of chambers, average
kv = 2.3570e-07;             % Orifice equation constant

```

```
%%  Flows are calculated
% Inflow to chambers due to ON valve connected to manifolds
QC1_in = u(1)*kv*sqrt(abs(pS-q(3)))* (q(3)<pS);
QC2_in = u(2)*kv*sqrt(abs(pS-q(4)))* (q(4)<pS);
QC3_in = u(3)*kv*sqrt(abs(pL-q(5)))* (q(5)<pL);
QC4_in = u(4)*kv*sqrt(abs(pL-q(6)))* (q(6)<pL);

% Outflow from chambers due to ON valve connected to tank
QC1_out = u(5)*kv*sqrt(abs(p0-q(3)))* (q(3)>p0);
QC2_out = u(6)*kv*sqrt(abs(p0-q(4)))* (q(4)>p0);
QC3_out = u(7)*kv*sqrt(abs(p0-q(5)))* (q(5)>p0);
QC4_out = u(8)*kv*sqrt(abs(p0-q(6)))* (q(6)>p0);

% Inflow to chambers due to CHECK valves connected to tank
QC1_checkin = kv*sqrt(abs(p0-q(3))) * (q(3)<p0);
QC2_checkin = kv*sqrt(abs(p0-q(4))) * (q(4)<p0);
QC3_checkin = kv*sqrt(abs(p0-q(5))) * (q(5)<p0);
QC4_checkin = kv*sqrt(abs(p0-q(6))) * (q(6)<p0);

% Outflow from chambers due CHECK valve connected to manifold
QC1_checkout = kv*sqrt(abs(pS-q(3))) * (q(3)>pS);
QC2_checkout = kv*sqrt(abs(pS-q(4))) * (q(4)>pS);
QC3_checkout = kv*sqrt(abs(pL-q(5))) * (q(5)>pL);
QC4_checkout = kv*sqrt(abs(pL-q(6))) * (q(6)>pL);



%% Dynamic equations
qdot(1,1) = q(2);                                                        % speed
qdot(1,2) = (q(3)*A - q(4)*A + q(5)*A - q(6)*A - B*q(2))/m;              % acceleration
qdot(1,3) = beta/V * (QC1_in - QC1_out - A*q(2) + QC1_checkin - QC1_checkout);  % Continuity chamber 1
qdot(1,4) = beta/V * (QC2_in - QC2_out + A*q(2) + QC2_checkin - QC2_checkout);  % Continuity chamber 2
qdot(1,5) = beta/V * (QC3_in - QC3_out - A*q(2) + QC3_checkin - QC3_checkout);  % Continuity chamber 3
qdot(1,6) = beta/V * (QC4_in - QC4_out + A*q(2) + QC4_checkin - QC4_checkout);  % Continuity chamber 4
```

E.11 Plant Reduced Model

```
    function qdot = SystemPLANT(q,u,params)
%%% System_reduced %%%
% This system is called PLANT, since I want the possibility to make this
% different than the model. Could include disturbances and such.
% But for now is the same as the model used in optimization.

%% Getting constants - parameters

p0 = params(5); %
pS = params(6); %
pL = params(7); %

```

```
% System constants are loaded
A = (1.22e-04)/2;           % Piston area
m = 1101.79e-3+417.41e-3;   % mass piston
B = params(8);              % Damping term
beta = 1.4000e+09;          % Bulk modulus. We could have term for evaluating bulk
    modulus for current pressures.
V = 2.6996e-05;             % Volume of chambers, average
kv = 2.3570e-07;            % Orifice equation constant


%%  Flows are calculated
% Inflow to chambers due to ON valve connected to manifolds
QC1_in = u(1)*kv*sqrt(abs(pS-q(3)))* (q(3)<pS);
QC2_in = u(2)*kv*sqrt(abs(pS-q(4)))* (q(4)<pS);
QC3_in = u(3)*kv*sqrt(abs(pL-q(5)))* (q(5)<pL);
QC4_in = u(4)*kv*sqrt(abs(pL-q(6)))* (q(6)<pL);

% Outflow from chambers due to ON valve connected to tank
QC1_out = u(5)*kv*sqrt(abs(p0-q(3)))* (q(3)>p0);
QC2_out = u(6)*kv*sqrt(abs(p0-q(4)))* (q(4)>p0);
QC3_out = u(7)*kv*sqrt(abs(p0-q(5)))* (q(5)>p0);
QC4_out = u(8)*kv*sqrt(abs(p0-q(6)))* (q(6)>p0);

% Outflow from chambers due CHECK valve connected to manifold
QC1_checkout = kv*sqrt(abs(pS-q(3))) * (q(3)>pS);
QC2_checkout = kv*sqrt(abs(pS-q(4))) * (q(4)>pS);
QC3_checkout = kv*sqrt(abs(pL-q(5))) * (q(5)>pL);
QC4_checkout = kv*sqrt(abs(pL-q(6))) * (q(6)>pL);

% Inflow to chambers due to CHECK valves connected to tank
QC1_checkin = kv*sqrt(abs(p0-q(3))) * (q(3)<p0);
QC2_checkin = kv*sqrt(abs(p0-q(4))) * (q(4)<p0);
QC3_checkin = kv*sqrt(abs(p0-q(5))) * (q(5)<p0);
QC4_checkin = kv*sqrt(abs(p0-q(6))) * (q(6)<p0);

%% Dynamic equations
qdot(1,1) = q(2);                                                          %
    speed
qdot(1,2) = (q(3)*A - q(4)*A + q(5)*A - q(6)*A - B*q(2))/m;                %
    acceleration
qdot(1,3) = beta/V * (QC1_in - QC1_out - A*q(2) + QC1_checkin - QC1_checkout);  %
    Continuity chamber 1
qdot(1,4) = beta/V * (QC2_in - QC2_out + A*q(2) + QC2_checkin - QC2_checkout);  %
    Continuity chamber 2
qdot(1,5) = beta/V * (QC3_in - QC3_out - A*q(2) + QC3_checkin - QC3_checkout);  %
    Continuity chamber 3
qdot(1,6) = beta/V * (QC4_in - QC4_out + A*q(2) + QC4_checkin - QC4_checkout);  %
    Continuity chamber 4
```

E.12 Input Matrix Generator from Switching Times

```
    function U = tswitch2inputs(xdes_pool,params)
% This function generates a matrix (p X nu) of inputs based on the design
% variables i.e. switching times
xdes = xdes_pool;
```

```
%% Loading parameters
    Ts       = params(1);
    p        = params(2);

%% Sorting switching times into a matrix
    t_switch        = zeros(2,8);
    t_switch(1,1) = xdes(1);
    t_switch(2,1) = xdes(2);
    t_switch(1,2) = xdes(3);
    t_switch(2,2) = xdes(4);

    t_switch(1,3) = xdes(5);
    t_switch(2,3) = xdes(6);
    t_switch(1,4) = xdes(7);
    t_switch(2,4) = xdes(8);

    t_switch(1,5) = xdes(9);
    t_switch(2,5) = xdes(10);
    t_switch(1,6) = xdes(11);
    t_switch(2,6) = xdes(12);

    t_switch(1,7) = xdes(13);
    t_switch(2,7) = xdes(14);
    t_switch(1,8) = xdes(15);
    t_switch(2,8) = xdes(16);

    for i = 1:8
        if t_switch(1,i) < Ts;         t_switch(1,i) = Ts;   end
        if t_switch(2,i) < Ts;         t_switch(2,i) = Ts;   end
        if t_switch(1,i) > Ts*p;       t_switch(1,i) = Ts*p; end
        if t_switch(2,i) > Ts*p;       t_switch(2,i) = Ts*p; end
    end

    nT = round(t_switch/Ts); % time samples
    for j = 1:8
        if nT(1,j) == 2
            nT(1,j) = 1;
        end
    end

%% Generating Input matrix
    U = zeros(p,8);                    % Initializing input matrix
    U(:,5:8) = 1;                      % inputs u5 and u6 (LP C1 and C2) are on as default
                                       % If switching times are equal, there should not be generated input
                                       % Indices corresponding to switching times are made as positive values
    for i = 1:4
            if ~(nT(1,i) == nT(2,i))
            U(nT(1,i):nT(2,i) ,i)   = 1; % u1 to u4 are turned on
            end
    end
    for i = 5:8
            if ~(nT(1,i) == nT(2,i))
            U(nT(1,i):nT(2,i) ,i)   = 0; % u5 to u8 are turned off
            end
```

```
    end
end
```

E.13 Work Evaluator

```
    function [WO, WI] = Work(A_set,q0,q_old,WO,WI)
deltax = q0(1) - q_old(1);
PresI = (q0(3) - q0(4));
PresO = (-q0(5) + q0(6));

WIs = deltax*PresI*A_set;
WOs = deltax*PresO*A_set;
WI = WI + WIs;
WO = WO + WOs;
```

E.14 Switching Time Generator

```
    function xdes_pool = xdes2pool(xdes,params)
% This function generates switching times based on design variables
% Loading parameters
Ts          = params(1);
p           = params(2);
Pump        = params(10);
xref        = params(4);
target      = xref > 17.5*10^-3;
trigger     = params(12:13);

tdelay      = 1*10^-3; % A purposeful delay between turning off LP and turning on HP,
                  % such that pressure can be built up

xdes_pool   = zeros(16,1); % Initialize switching times

[xIHP, xILP, xOLP, xBHP, xBLP] = Cases(Pump,target); % Identify switching time roles

xdes_pool(xIHP(1)) = Ts;        % Turn on Input HP at beginning of pred. horizon. bc. there is no reason to turn it on at a later time
xdes_pool(xIHP(2)) = xdes(1);   % Inserting design value at correct place in pool. Input HP turns off
xdes_pool(xILP(1)) = Ts;        % Turn off Input LP at beg. p. hor.: So create HP. Otherwise HP to tank = bad
xdes_pool(xILP(2)) = p*Ts; % Turn on Input LP at end of prediction horizon. There is no reason to turn on before this

xdes_pool(xOLP(1)) = xdes(1);   % Turn off Output LP when turning off Input HP. So output pressure can build up
xdes_pool(xOLP(2)) = p*Ts;      % Turn off Output LP at end of pred. horizon to maximize output potential

xdes_pool(xBLP(1)) = xdes(2)-tdelay; % Turn off braking LP when turning on braking HP
xdes_pool(xBLP(2)) = p*Ts;      % Turn on braking LP at end of. hor. Same argument as above.
```

```
xdes_pool(xBHP(1)) = xdes(2);   % If not triggered, then the braking HP turn on is
    as specified by designvariable
xdes_pool(xBHP(2)) = p*Ts;      % Turn off braking HP at end of p. hor. It should
    never end before that, since the pressure should be high for next half-cycle

if trigger(1) == 2 % If the input has been turned off after being on, it must not be
     turned on again. Regarding Input
    xdes_pool(xIHP(2))    = Ts;  % The Input turn off time is set to zero essentially
    xdes_pool(xILP(2))    = Ts;  % Input LP turn on is set to zero time = immediate
        activate so Input pressure does not build up
    xdes_pool(xOLP(1))    = Ts;  % Turn off Output LP when turning off Input HP. So
        output pressure can build up
end
if trigger(2) == 2 % If the input has been turned off after being on, it must not be
     turned on again. Regarding Braking
    xdes_pool(xBHP(1))   = p*Ts; % The designvariable corresponding to turning on HP
        braking pressure is set to end of prediction horizon, so never actually turn
         on
    xdes_pool(xBLP(1))   = p*Ts; % Braking LP turn off is set to end of prediction
        horizon, so it essentially always is on. Pressure cannot build up
end

% Trim switching times if they are outside span of horizon
for i = 1:16
    if xdes_pool(i) < Ts;        xdes_pool(i) = Ts;   end
    if xdes_pool(i) > Ts*p;      xdes_pool(i) = Ts*p; end
end
```