UNIVERSITY OF AGDER

Secondary Control of a Digital Hydraulic Motor for Winch Applications

Maxime Marien
Kjell Erik Wiig

Supervisor
Morten Kjeld Ebbesen

This master's thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn

University of Agder, 2018
Faculty of Engineering and Science
Department of Engineering Sciences

Abstract

The reliable traditional hydraulic systems for industrial purposes is being challenged by an emerging technology called digital hydraulics. This technology shows promising improvements in regards to efficiency, especially in comparison to systems dependent on hydraulic pumps or motors with variable displacement. Digital hydraulic motors eliminate the need for a servo valve by connecting each individual motor cylinder directly to the high-pressure and low-pressure line by means of fast-acting on/off-valves.

In this thesis, the potential of digital hydraulics is investigated by developing a control system intended for a digital displacement motor used on an industrial winch application. The expected operation modes of the winch system are analysed, and different control strategies are discussed. The developed control system is tested for the expected operation modes before it is simulated with a model of an industrial winch.

The digital displacement motor performs well during simulated testing of typical load cases for a winch system. Some challenges are discovered when the motor switches operation mode and when the reference velocity is zero.

Solutions for counteracting the mentioned challenges are implemented, and further possible improvements are proposed. The presented results prove that the suggested control strategy has the potential for good reference tracking and that the overall solution is suitable for a winch system.

Preface

Digital Hydraulics is a relatively new field of study where most of the literature is found in academic research papers or theses, which can be difficult to navigate through without the correct guidance. The authors would like to thank PhD Research Fellow Sondre Nordås for pointing out direction through the jungle of information that exists on this subject, sharing of intricate figures and animations, and for great advice on all aspects of the project.

The authors would also like to thank Associate Professor Morten Kjeld Ebbesen for counselling and support throughout the project.

Maxime Marien   Kjell Erik Wiig

Maxime Marien   Kjell Erik Wiig

Grimstad, May 2018

Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	**Explanation**
HPU	Hydraulic Pressure Unit
TDC	Piston Top Dead Center
BDC	Piston Bottom Dead Center
HPV	High-pressure valve
LPV	Low-pressure valve
cc	Cubic Centimeter
DDM	Digital Displacement Motor
FSS	Full Stroke Displacement Strategy
PSS	Partial Stroke Displacement Strategy

Nomenclature

CHAPTER 2		
Variable	**Unit**	**Explanation**
p_H	Pa	Pressure in high-pressure manifold
p_L	Pa	Pressure in low-pressure manifold
Q_H	m^3/s	Flow in high-pressure valve
Q_L	m^3/s	Flow in low-pressure valve
$\bar{x}_p$	$-$	Normalized piston position , $x_p \in [0,1]$
$\theta_m, \dot{\theta}_m, \ddot{\theta}_m$	$rad, rad/s, rad/s^2$	Motor shaft angular position, velocity, acceleration
e	m	Motor shaft eccentricity
l_1	m	Slider-crank member 1
l_2	m	Slider-crank member 2
ϕ_1	rad	Slider-crank angle 1
ϕ_2	rad	Slider-crank angle 2
$x_{diff,max}$	mm	Maximum error slider-crank approximation
$p_c, \dot{p_c}$	$Pa, Pa/s$	Cylinder chamber pressure and pressure gradient
$V_c(\theta)$	m^3	Cylinder chamber volume as a function of shaft angle
V_0	m^3	Cylinder chamber dead volume
$\Delta\theta$	rad	Cylinder phase shift
n_c	-	Number of cylinders
$\theta_c, \dot{\theta}_c$	$rad, rad/s$	Angular position and velocity for each individual cylinder
i	-	Cylinder number
V_d	m^3	Displacement volume
A_p	m^2	Piston area
$\bar{x}_H, \bar{x}_L$	$-$	Normalized HP and LP valve positions, $x \in [0,1]$
k_f	$(\sqrt{Pa} \cdot s)/m^3$	Valve flow-pressure coefficient
t_s	s	Switching time
$\bar{t}_s$	-	Normalised switching time
$\Delta p_{valve,max}$	Pa	Highest pressure drop over the valve
Q_{max}	m^3/s	Highest expected flow over the valve
Q_{avg}	m^3/s	Average cylinder flow
Q_c	m^3/s	Cylinder flow
n_m	RPM	Motor shaft rotational velocity
β	Pa	Oil bulk modulus
T	K	Temperature
ϵ_g	$-$	fraction of entrapped gases
β_L	Pa	Bulk modulus of the liquid
p^{abs}	Pa	Absolute pressure
ϵ_{g0}	$-$	Fraction of entrapped gases at atmospheric pressure
p^{abs}_{atm}	Pa	Atmospheric pressure, absolute
κ	$-$	Adiabatic coefficient

CHAPTER 2 continued

Variable	**Unit**	**Explanation**
Q_{in}	m^3/s	Flow in to the control volume
Q_{out}	m^3/s	Flow out of the control volume
T_c	Nm	Torque contribution from one cylinder
T_m	Nm	Motor shaft resultant torque
θ_1	rad	First sequence angle
θ_2	rad	Second sequence angle
θ_3	rad	Third sequence angle
θ_4	rad	Fourth sequence angle
V_1	m^3	Volume at angle 2
V_2	m^3	Volume at angle 3

CHAPTER 3

Variable	**Unit**	**Explanation**
T_{avg}	Nm	Average Torque
α	rad	Displacement parameter
α_{max}	rad	Maximum value of the displacement parameter
$\dot{\theta}_{ref}$	rad/s	Velocity reference
α_{FF}	rad	Feed forward element
$\dot{z}_{pl}, z_{pl}, \ddot{z}_{pl}$	m/s	Payload position, velocity, acceleration

CHAPTER 4		
Variable	**Unit**	**Explanation**
d_d	m	Drum diameter
$L_{w,d}$	m	Length of wire on drum
w_d	m	Width of drum
d_w	m	Diameter wire
ρ_w	kg/m	Density wire
J_d	$kg \cdot m^2$	Drum inertia
m_w	kg	Mass of wire
k_w	N/m	Wire stiffness
b_w	$N \cdot s/m$	Wire damping
$\delta, \dot{delta}$	$m, m/s$	Wire elongation and elongation rate
m_{pl}	kg	Mass of payload
F_w	N	Force from wire
E	Pa	E-modulus
A	m^2	Wire cross-sectional area
L	m	Length of wire
r_d	m	Radius of drum
δ_0	m	Initial wire elongation
$F_{B,pl}$	N	Payload buoyancy
$F_{B,w}$	N	Wire buoyancy
g	m/s^2	gravitational acceleration
T_{visc}	Nm	Viscous friction moment
F_d	N	Force on drum
J_{eff}	$kg \cdot m^2$	Effective inertia
b_b	$Nm \cdot s$	Friction coefficient bearings
r_{eff}	m	Effective radius
n_l	$-$	Number of layers
n_t	$-$	Number of turns
$L_{w,t}$	$-$	Length of wire per turn
$E_{kin,tot}$	J	Total kinetic energy
J_m	$kg \cdot m^2$	Motor inertia
ω_d	s^{-1}	Drum velocity
$J_{w,d}$	$kg \cdot m^2$	Rotational inertia of wire in drum
$m_{w,d}$	kg	Mass of wire on drum
$T_{m,max}$	Nm	Maximum motor torque
$\ddot{\theta}_{m,dem}$	s^{-2}	Demanded acceleration
$T_{L,max}$	Nm	Maximum recorded load torque
V_{pl}	m^3	Volume payload
$\dot{z}_{pl,max}$	$m/2$	Maximum payload velocity
T_w	s	Wave period
A_w	m	Wave amplitude
$z_w, \dot{z}_w, \ddot{z}_w$	$m, m/s, m/s^2$	Wave position, velocity, acceleration
T_{dem}	Nm	Demanded torque

CHAPTER 5

Variable	Unit	Explanation
D_m	cm^3/rev	Total motor displacement
Δp_{motor}	Pa	Pressure drop over the motor
n_v	$-$	Number of valves
$\bar{t}_{s,dem}$	$-$	Requirement for normalised switch time
t_{charge}	s	Charge time
$\vec{p}$	$-$	polynomial vector

CHAPTER 6

Variable	Unit	Explanation
K_p	$-$	Proportional gain
T_i	s	Integral time constant
T_d	s	Derivative time constant

1 Introduction

This thesis proposes a control strategy for the secondary control of a digital hydraulic motor. In this thesis, digital hydraulics is referred to a motor/pump where the indivudal cylinders are directly connected to the high- and low-pressure supply lines through the means of fast acting on/off valves. The control of the unit is thus implemented directly on the motor, also called the secondary unit (the primary unit being the pump/HPU that delivers the hydraulic system pressure), and not through a conventionally used servo valve.

The goal is to reduce the power loss often associated with a traditional variable displacement motor, especially at low displacement. The developed control system uses a displacement strategy that enables adjustment of the motor displacement while maintaining high system efficiency. In this thesis, this is referred to as Partial Stroke Displacement.

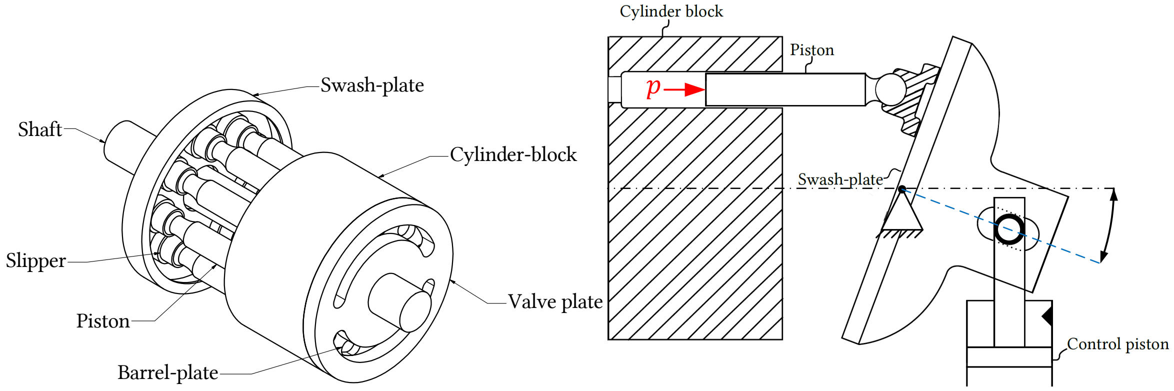
To test the control concept, a traditional winch setup is modelled and connected to the motor in a simulations. Different velocity reference profiles are applied to digital motor, in order to test the handling ability of light and heavy payloads.

1.1 Background and Motivation

Hydraulic fluid power systems are often used as actuators for industrial purposes because of their high power-to-density ratio compared to other alternatives such as electrical drive trains [3]. Various types of motors exists for different applications. Clearly, the used industry standard for hydraulic machines is well-proven and robust. However, the traditional working principles are challenged due to recent demands regarding power consumption and efficiency, confirmed by leading industry brands [4]. It is therefore desirable to modernize the hydraulic pump/motor in order to meet these requirements. This section will shortly present one commonly used combination of hydraulic motor and winch, and a state-of-the-art for the digital displacement motor to clarify the motivation behind digital displacement motors.

1.1.1 Variable Displacement Axial Piston Motor

A typical motor configuration for applications requiring a variable displacement motor is the axial piston motor. In essence, pressurized oil displaces a set of pistons translating in parallel direction to the shaft, hence the name *axial piston machine*. An example of such machine is presented in figure 1.1a. High efficiency is obtained during operation at displacements of 90% or greater. However, at lower displacement fractions, the efficiency drops significantly to 60-80% [1].



(a) Axial Piston Machine Overview (b) Displacement Adjustment

Figure 1.1: Variable displacement axial piston motor

Figure 1.1b shows that by actuating the control piston, adjustment of the swash-plate angle is achieved, which again changes the stroke length travelled by the pistons and thereby the displacement of the motor. Each cylinder chamber is pressurized regardless of the selected displacement, which is considered one of the main disadvantages with this design [5]. Pressurization of the cylinder chamber generates a force acting normal on the swash-plate, which also have a radial component. Consequently, a friction force between the piston and the cylinder block is generated, which degrades the motor hydro-mechanical efficiency. Also, pressurized cylinders induces leakage over the pistons, reducing the volumetric efficiency [1].

For the axial piston machine to be suitable for heavy-duty winch configurations, additional reduction gear is often required to raise the torque level and lower the output speed, thus degrading the efficiency. In addition, the added gearbox may also require a cooling system, which further increases the cost and reduces the efficiency of the total system.

1.1.2 State-of-the-Art Digital Displacement Motors

The core concept of a digital displacement motor (DDM) is controlling the motor cylinders independently from one another, using fast-switching digital ON/OFF valves. Figure 1.2 shows a typical configuration for a DDM. The DDM is built up in a radial-piston configuration, evenly distributed around an eccentric cam shaft. Each cylinder is individually connected to the high-pressure line and the low-pressure line by two fast-switching on/off-valves, HPV (high pressure valve) and LPV (low pressure valve) respectively.

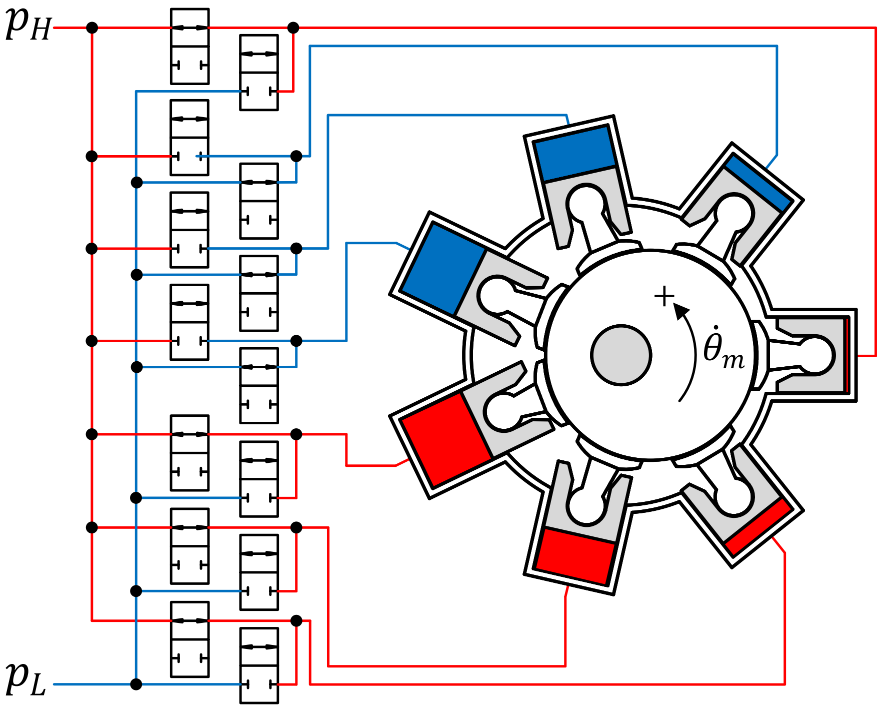


Figure 1.2: DDM schematic overview

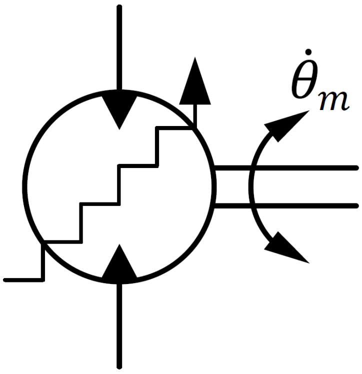


Figure 1.3: DDM Displacement Discretization

Blue coloured lines and cylinder chambers represent low pressure fluid, and the red ones represent high pressure fluid. The modular-based design of the DDM makes it possible to increase the motor size by simply adding cylinder rows. Motor maintenance is also facilitated as the individual cylinders and respective valves are easily accessible. As the name implies, discretisation of the displacement lies in the nature of this machine, due to the common used displacement strategy of activating or de-activating each cylinder during one cycle. De-activation of a cylinder is achieved by simply closing the valve connected to the high-pressure line and opening the valve connected to the low-pressure line, allowing the piston to passively follow the movement of the rest of the motor while displacing low pressure fluid. Figure 1.3 illustrates the discretisation of the variable displacement.

Development of this type of hydraulic actuactors has emerged over the last couple of decades. The technology is still mostly in the experimental phase, with a handful of commercial companies developing own prototypes. Preliminary studies show promising results in regards to the efficiency compared to traditional variable displacement hydraulic motors. In contrast to axial piston machines where all cylinders are pressurised, the deactivated cylinders are not pressurized, which reduces internal friction and leakages and increases efficiency. A DDM can reach efficiencies as high as 97% at full displacement, and 96% at 20% displacement, shown in figure 1.4, which clearly states the motivation behind this technology. Also, replacing a typical servo valve setup with a set of practically leakage-free seat valves on every cylinder reduces leakages and further increases system efficency.

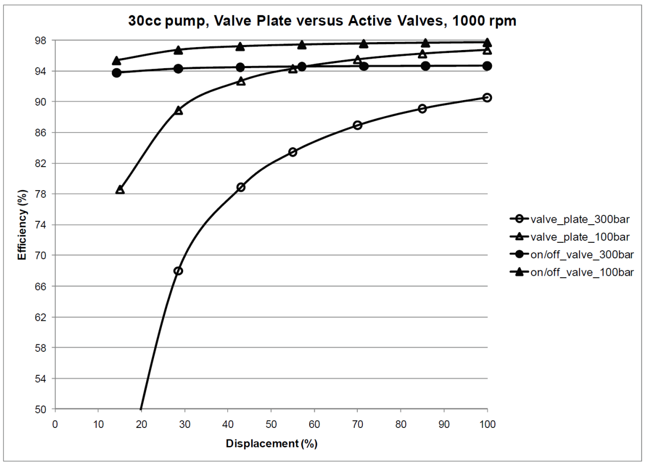


Figure 1.4: Efficieny as a function of displacement, digital displacement pump versus axial piston pump [1]

Artemis Intelligent Power Ltd. was formed in 1994, and is by now considered one of the leading companies within the next generation of hydraulic machines using digital displacement technology. Tests performed on their highly efficient digital machines showed a great reduction of heat generation compared to a conventional axial piston machine. Thus, costs can be reduced when considering cooling equipment, or by reducing the power requirements of the hydraulic power unit (HPU) [6].

1.2 Problem Statement

This section presents the problem statement defined for this thesis.

A control system for a digital displacement motor is to be designed and tested on a model of a typical industrial winch setup. Important design criteria for the control system include:

- Analysis of applicable displacement strategies
- Secondary control of a high torque low speed digital displacement motor

The winch setup represents a typical real world industrial case based on a 20 tonne hydraulic winch and should incorporate the necessary mechanical and hydraulic details needed to reach a reasonable modelling accuracy.

The system simulation model should be able to do the following:

- Hoist and lower load
- Steady handling of both heavy and light load
- Follow an angular velocity reference, both unidirectional and varying (simulating a traditional heave compensation movement)

2 Fundamentals of Digital Displacement Motors

This chapter presents the working principles of a digital displacement motor, and discusses key aspects concerning modelling of the motor dynamics and the digital valves. A general procedure for correct timing of the individual cylinder valves is also presented. The presented composition of the radial piston motor is simplified to some extent, compared to a full mechanical model including friction and internal leakage. This is due to the attention directed to the development of a control system.

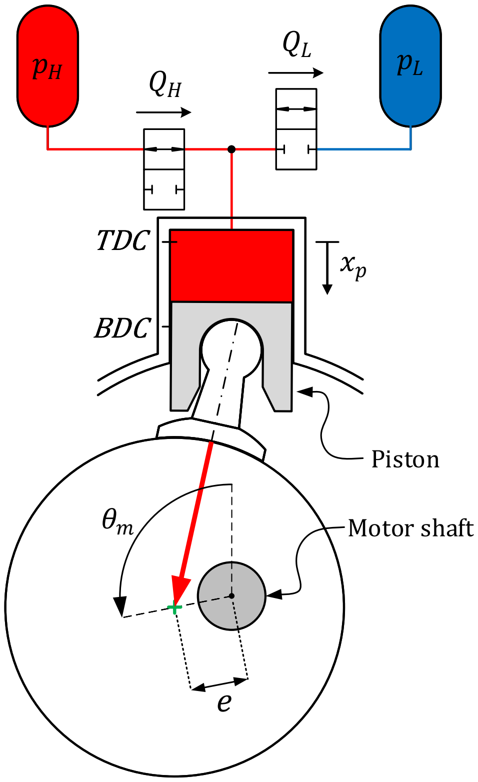


Figure 2.1: Single Cylinder DDM

Figure 2.1 shows a simplified model of a single cylinder DDM with a description of some of the key components and parameters. The high-pressure supply p_H is shown in red, and the low-pressure side p_L is shown in blue. The HPV is shown in open position and the LPV in closed positon. The flow over the valves Q_H (through the HPV) and Q_L (through the LPV) are shown in positive direction. The piston travel is denoted x_p, and is defined as zero at the cylinder TDC (Top Dead Centre). The cylinder chamber is pressurised due to the open HPV, which translates into a force acting from the piston on to the eccentric moment arm, e. Thus, the force (indicated by the red arrow), applies a torque to the motor shaft. This torque causes the shaft to rotate. The motor angle is denoted θ_m. The details of these individual aspects are described in depth in the subsequent sections.

2.1 Piston Kinematics

This section discusses the kinematics of the cylinder piston, and how it can be simplified in order to reduce modelling complexity. For a DDM with radial piston configuration, each cylinder behaves as a slider-crank mechanism. Figure 2.2 shows a typical slider-crank setup, where the piston translates horizontally as the angle ϕ_1 rotates.

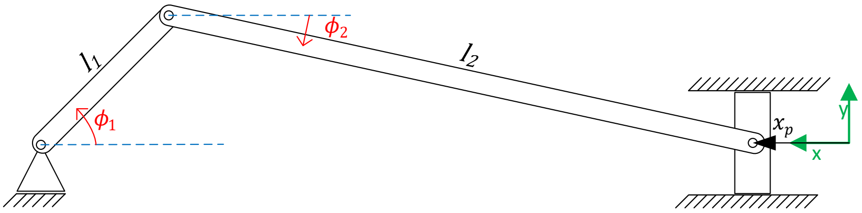


Figure 2.2: Slider-crank mechanism

The expression for the piston position can then be derived from equation 2.1:

$$x_p = l_1 - l_1 \cdot \cos(\phi_1) + l_2 - l_2 \cdot \cos(\phi_2) \tag{2.1}$$

Using the physical constraint that the difference in y-direction must be zero, the angle ϕ_2 can be expressed as a function of ϕ_1, reducing the number of variables:

$$l_1 \cdot \sin(\phi_1) - l_2 \cdot \sin(\phi_2) = 0$$
$$\Downarrow$$
$$\phi_2 = \sin^{-1}\left(\frac{l_1}{l_2} \cdot \sin(\phi_1)\right) \tag{2.2}$$

In a conventional radial piston motor, l_2 is relatively greater than l_1. This relation greatly affects the maximum value of the angle ϕ_2. Small values of ϕ_2 yields $l_2 - l_2 \cdot \cos(\phi_2) \approx 0$. A simpler expression is therefore set up with reduced complexity. To relate the slider-crank setup to the DDM shown in figure 2.1, the parameter l_1 is replaced with the motor shaft eccentricity e, and the angle ϕ_2 is replaced with the motor angle θ_m.

$$x_p = e - e \cdot \cos(\theta_m) \tag{2.3}$$

Figure 2.3 plots the two piston expressions with arbitrary values ($l_1 = 10$ mm and $l_2 = 10 \cdot l_1$).

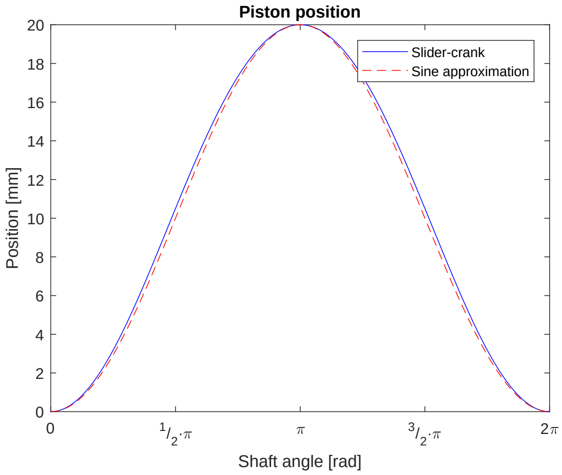


Figure 2.3: Slider-crank versus sine approximation

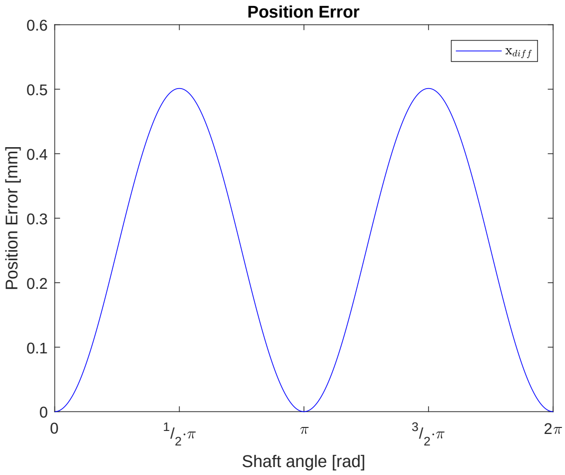


Figure 2.4: Acceptable error

Figure 2.4 shows an acceptable accuracy trade-off ($x_{diff,max} = 0.5mm$) for the simpler expression. Equation 2.3 is therefore further used in this project work to represent the piston movement.

2.2 Inherent Characteristics

This section presents aspects of the DDM that are needed to derive expressions used in the modelling of the motor. The single cylinder motor, figure 2.5, is again used to determine motor parameters:

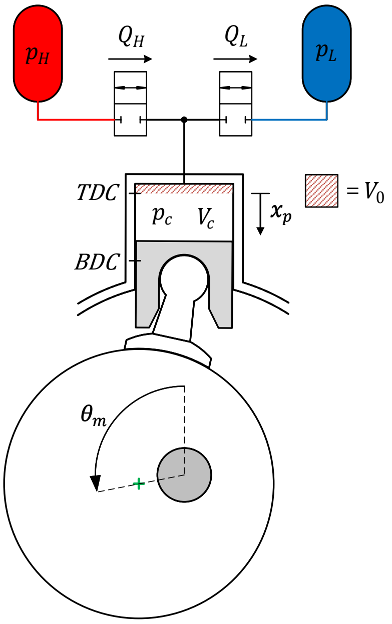


Figure 2.5: Single Cylinder DDM

- The cylinder has a dead volume, denoted V_0
- The total cylinder volume including the dead volume is denoted V_c
- The pressure in the cylinder chamber is denoted p_c
- The flow through the HPV Q_H is defined as positive going from the high-pressure line to the cylinder chamber
- The flow through the LPV Q_L is defined as positive going from the cylinder chamber to the low-pressure line
- The motor shaft angle is defined as positive in the CCW direciton

Expression for the Individual Cylinder Angle

When considering a more realistic motor with multiple cylinders equally spaced around the motor, it can be useful to define an angle for each individual cylinder. Figure 2.6 shows a multiple-cylinder motor with equally spaced cylinders, with an angular difference of $\Delta\theta$:

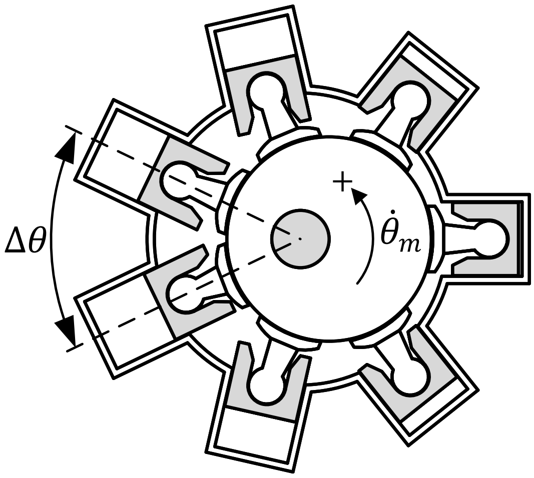


Figure 2.6: Multiple Cylinder DDM

The expression for $\Delta\theta$ must be

$$\Delta\theta = \frac{2\pi}{n_c} \tag{2.4}$$

where n_c is the number of cylinders. The expression for the cylinder angle must be defined so that it is equal to $n \cdot 2\pi$ at each cylinder's respective TDC, where n is any integer. The expression per cylinder is then

$$\theta_c = \theta_m - \Delta\theta \cdot (i - 1) \tag{2.5}$$

where θ_c is the cylinder angle, θ_m the motor angle and i the cylinder number. It then follows that $\dot{\theta}_c = \dot{\theta}_m$, as the only difference between the two is a phase shift.

Volume Expression

The motor can be modelled with a single geometrical parameter, the total displaceable cylinder volume:

$$V_d = 2 \cdot A_p \cdot e \tag{2.6}$$

where V_d is the total displaceable volume, A_p the piston area and e the shaft eccentricity. This corresponds to a piston position at BDC (Bottom Dead Centre), where the cylinder volume is largest. Together with the simplified expression for the piston movement (equation 2.3), the cylinder volume V_c can then be expressed as:

$$\begin{aligned} V_c &= V_0 + A_p \cdot x_p \\ V_c &= V_0 + \frac{V_d}{2} \left(\cdot 1 - \cos\left(\theta_c\right)\right) \end{aligned} \tag{2.7}$$

where V_0 is the cylinder dead volume, V_d is the total displaceable volume and θ_c the cylinder angle. The rate of change of the cylinder volume is then the derivative of equation 2.7:

$$\dot{V}_c = \frac{V_d}{2} \cdot \dot{\theta}_m \cdot \sin(\theta_c) \tag{2.8}$$

where $\dot{V}_c$ is the rate of change of the cylinder volume and $\dot{\theta}_m$ the shaft angular velocity.

2.3 Digital Valves

In this thesis, the term digital valve is used of a seat valve typically actuated by a solenoid (some featuring pilot pressure assistance) that has only two possible configurations, fully open and fully closed. These valves, also called on/off valves, are characterised by two main parameters; the flow-pressure coefficient k_f, and the valve switching time t_s. The relationship between the flow over the valve and the pressure difference is described by the simplified orifice equation 2.9:

$$Q = \frac{\bar{x}}{k_f}\sqrt{\Delta p_{valve}} \tag{2.9}$$

where Q is the flow through the valve, $\bar{x}$ the normalized plunger position, Δp_{valve} the pressure difference over the valve and k_f the flow-pressure coefficient. For an ideal valve, it follows that the flow-pressure coefficient should have a low value for the valve to deliver sufficient flow at a low pressure difference.

A low switching time is also desirable, in order to reduce the impact of the valve dynamics on the system efficiency. However, it can be shown that it is sufficient to ensure a low switching time relative to the operational speed of the motor. A normalized switching time of less than 10% of the motor rotation period is recommended, and less than 2-3% is desirable [7, p. 46]. These low percentages are considered achievable for low operational speeds.

The detailed modelling of these valves are outside the scope of this project. Instead, a simple model is used for describing the translational behaviour of the valve plunger, where the plunger moves from one position to the other with a constant acceleration phase and a constant deceleration phase. Velocity and position is then simply obtained by time-integration, see figure 2.7.

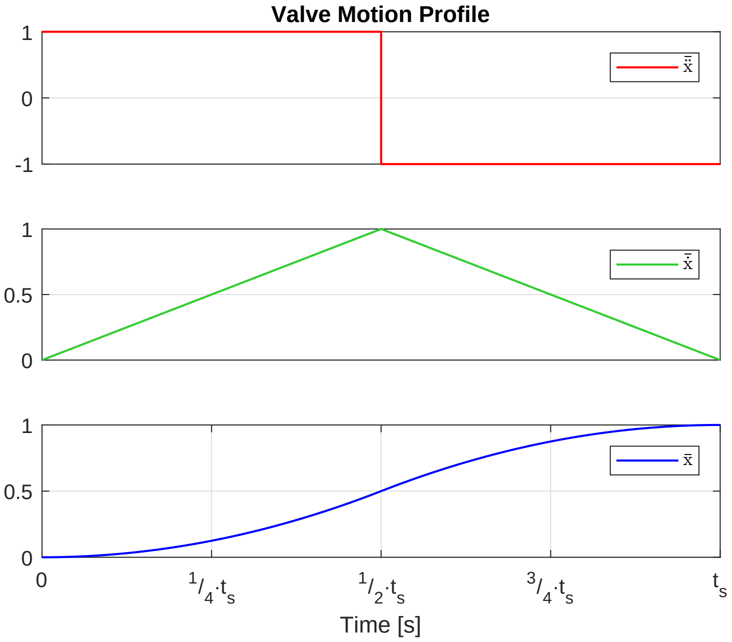


Figure 2.7: Generic motion profile based on switching time

In addition, the valves are modelled with a charge time t_{charge} that starts as a closing/opening sequence ends. During this small period, the valve can not take a new action. This represents the time needed for the valve actuator to prepare for a new actuation sequence, e.g. the build-up of a solenoid voltage.

In order to describe the flow through the digital valves, conventional orifice equations are used. The full version of orifice equation 2.9 is used to define oil flow through the respective cylinder high- and low-pressure valve. Normalized valve plunger position, $\bar{x} \in [0, 1]$, is utilized as indicated in figure 2.7.

$$Q_H = \frac{\bar{x}_H}{k_f} \cdot \sqrt{|p_H - p_c|} \cdot sign(p_H - p_c) \tag{2.10}$$

$$Q_L = \frac{\bar{x}_L}{k_f} \cdot \sqrt{|p_c - p_L|} \cdot sign(p_c - p_L) \tag{2.11}$$

where

Q_H	-	Flow through high-pressure valve
Q_L	-	Flow through low-pressure valve
$\bar{x}_H, \bar{x}_L$	-	Normalized HP and LP valve plunger positions
p_H	-	Pressure in the high-pressure manifold
p_L	-	Pressure in the low-pressure manifold
p_c	-	Pressure in the cylinder chamber

As the pressure difference is under a root sign, the absolute value is taken and the sign is included to the equation in order to determine the direction of the oil flow.

2.3.1 Requirements for Valve Selection

In order to select appropriate valves for a motor, requirement specifications for the valve parameters must be expressed. As stated earlier in this section, a normalized switching time of $\bar{t}_s = 2-3\%$ is desirable. Therefore, the requirement for the switching time t_s is simply a fraction of the motor revolution period T_{rev} at the reference speed, where the fraction represents the normalised switching time:

$$t_s = \bar{t}_s \cdot T_{rev} \tag{2.12}$$

The flow-pressure coefficient k_f is determined using a maximum allowed pressure drop over the valve, and the maximum flow demand Q_{max}:

$$k_f = \frac{\sqrt{\Delta p_{valve,max}}}{Q_{max}} \tag{2.13}$$

To find $\Delta p_{valve,max}$, the efficiency chart from [2, p. 49] is used, see figure 2.8. The left hand charts show the valve efficiency as a function of the maximum allowed pressure drop over the valve, the pressure drop over the motor, and the normalized switching time for three different partial displacement fractions. The right hand graphs shows the motor utilization, i.e. the actual displacement of the motor relative to the desired displacement fraction. Desired values for the motor efficiency and valve switching time must be determined before consulting the graph and finding a pressure ratio that can be solved for $\Delta p_{valve,max}$.

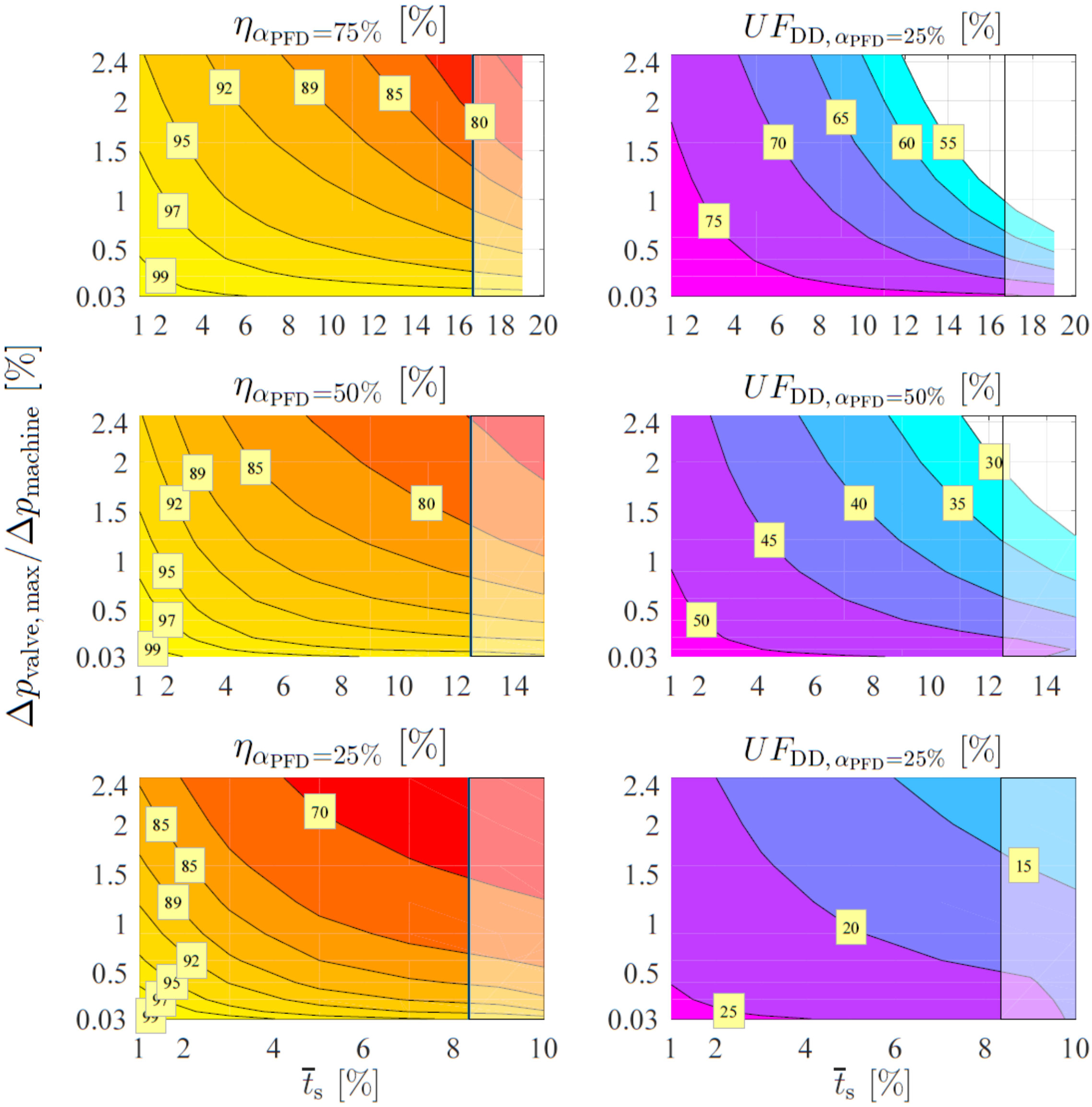


Figure 2.8: Valve efficiency (left) and motor utilization (right), at 25%, 50% and 75% displacement [2, p. 49]

Q_{max} can be found by investigating the cylinder's flow properties. Conventionally, the cylinder flow is computed as in equation 2.14:

$$Q_{avg} = V_d \cdot n_m \tag{2.14}$$

where Q_{avg} is the average cylinder flow, V_d is the displaceable cylinder volume, and n_m the motor rotational speed. However, when considering the featured radial piston design of a DDM, the requirement for flow supplied to the motor is altered due the way the machine displaces fluid. When considering an isolated cylinder, the chamber is pressurised when the piston travels from TDC to BDC, which corresponds to one half shaft revolution. Thus, the flow must be doubled compared to equation 2.14, indicated in figure 2.9:

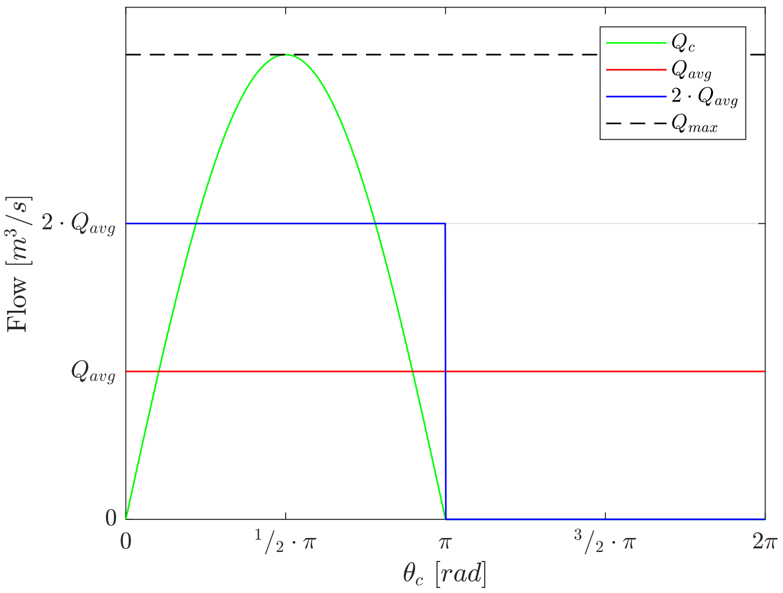


Figure 2.9: Relationship between flows

Further, the actual flow entering the cylinder chamber, Q_c, has a sinusoidal behaviour, and can thus be approximated by a sine function with an amplitude of Q_{max}. Over the piston stroke from TDC to BDC, the same amount of fluid must be displaced:

$$\int_0^\pi Q_{max} \cdot \sin(\theta_c)\ d\theta_c = \int_0^\pi 2 \cdot Q_{avg}\ d\theta_c \tag{2.15}$$

Thus, by solving this equation, the relationship between Q_{max} and Q_{avg} can be defined:

$$-Q_{max} \cdot [\cos(\theta_c)]\Big|_0^\pi = [2 \cdot Q_{avg} \cdot \theta_c]\Big|_0^\pi$$

$$Q_{max} = Q_{avg} \cdot \pi \tag{2.16}$$

With the maximum flow demand Q_{max} and $\Delta p_{valve,max}$ defined, equation 2.13 can be used to calculate the needed value of the flow-pressure coefficient k_f. The expressions presented in this section are only approximations of a normal piston stroke used to express the maximum flow demand for the valve used in equation 2.13. Therefore, it is important to notice that the actual flow into or out of the cylinder chamber is governed by the valve orifice equations (2.10 and 2.11).

2.4 Bulk Modulus

As the model of the DDM will include chamber pressure dynamics, the fluid bulk modulus should not be modelled as constant. The bulk modulus is a parameter describing the fluid stiffness, and can for simplicity be compared with a conventional spring constant. However, the bulk modulus is more complex, and depends on system pressure, temperature and the fraction of entrapped gasses in the fluid [8, Ch. 1.6], illustrated by equation 2.17. Figure 2.10 shows how the fraction of entrapped gasses influences the bulk modulus.

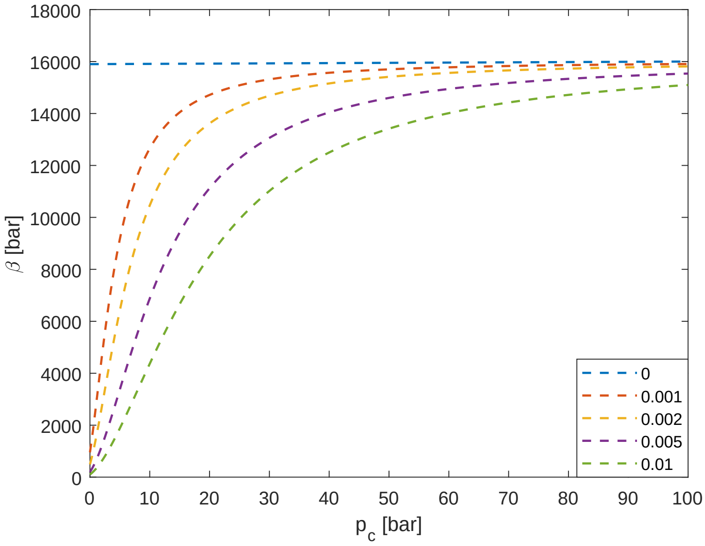


Figure 2.10: Change of oil stiffness depending on fraction of entrapped gases

$$\beta = \beta(p_c, T, \epsilon_g) \tag{2.17}$$

where

β	-	Bulk Modulus
p_c	-	Cylinder chamber pressure
T	-	Temperature
ϵ_g	-	Fraction of entrapped gases

As a consequence, an exact simulation of the bulk modulus requires a quite complex model, and is often simplified to some extent. In simple simulations, the bulk modulus is set constant, which may produce accurate results if the system pressure is sufficiently high. However, this assumption is not valid at lower system pressure, seen in figure 2.10.

In this project work, isothermal oil compression and decompression is assumed, i.e. the temperature is assumed constant leading to equation 2.18. Thus, the complexity is reduced while maintaining important properties.

$$\beta = \beta(p_c, \epsilon_g) \tag{2.18}$$

The bulk modulus is modelled as shown in equation 2.19, based on [8, p. 29]:

$$\beta = \frac{1}{\frac{1}{\beta_L} + \frac{\epsilon_g}{p^{abs}}} \tag{2.19}$$

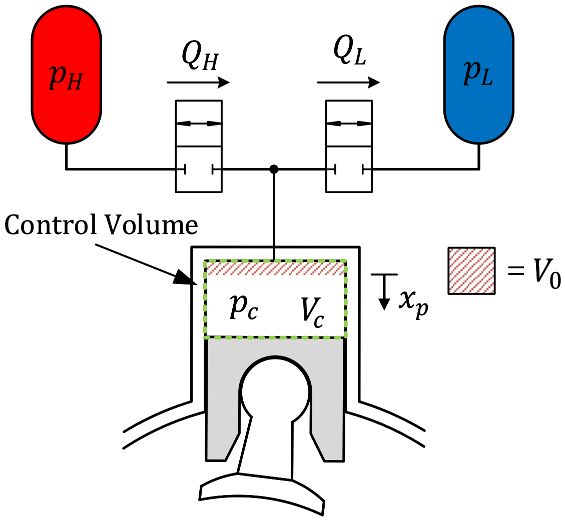
where β is the actual bulk modulus, β_L the liquid's bulk modulus, ϵ_g the fraction of entrapped gasses in the liquid and p^{abs} the absolute pressure in the liquid. The fraction of entrapped gasses also varies with the pressure in the liquid:

$$\epsilon_g = \frac{1}{\frac{1-\epsilon_{g0}}{\epsilon_{g0}} \cdot \frac{p_{atm}^{abs}}{p^{abs}}^{-\frac{1}{\kappa}} + 1} \tag{2.20}$$

where ϵ_{g0} is the fraction of entrapped gasses at atmospheric pressure, p_{atm}^{abs} is the absolute atmospheric pressure, p^{abs} the absolute pressure of the liquid and κ the adiabatic coefficient. The derived expressions in this section forms a dynamic bulk modulus which can be implemented in simulations.

2.5 Torque & Pressure Gradient

This section presents how the pressure in the cylinder is modelled, and subsequently the torque from the motor. A control volume is defined on the cylinder volume V_c, shown in figure 2.11, in order to apply the continuity equation (equation 2.21).



$$Q_{in} - Q_{out} = \dot{V} + \frac{V}{\beta} \cdot \dot{p} \tag{2.21}$$

Figure 2.11: Definition of control volume

The flows in and out of the control volume are Q_H and Q_L, from the HPV and LPV respectively. The expression for the rate of change of the control volume V_c has been determined in section 2.2 (equation 2.8). These values are inserted to the continuity equation which in turn is expressed for the pressure gradient $\dot{p_c}$:

$$\dot{p_c} = \frac{\beta}{V_c} \cdot (Q_H - Q_L - \dot{V_c}) \tag{2.22}$$

This expression is used to determine the pressure in the individual cylinder chambers p_c by time-integration in the simulation. As shown in figure 2.1 the cylinder pressure produces a torque on the motor shaft T_c, described by equation 2.23:

$$\begin{aligned} T_c &= p_c \cdot A_p \cdot e \cdot \sin(\theta_c) \\ T_c &= p_c \cdot \frac{V_d}{2} \cdot \sin(\theta_c) \end{aligned} \tag{2.23}$$

where A_p is the piston area, e the cam eccentricity, V_d the displaceable volume and θ_c the cylinder angle. Torque produced by the entire motor, T_m, is then simply the sum of all cylinder contributions, where n_c is the number of cylinders:

$$T_m = \sum_{i=1}^{n_c} T_i \tag{2.24}$$

2.6 Chamber Pressurisation

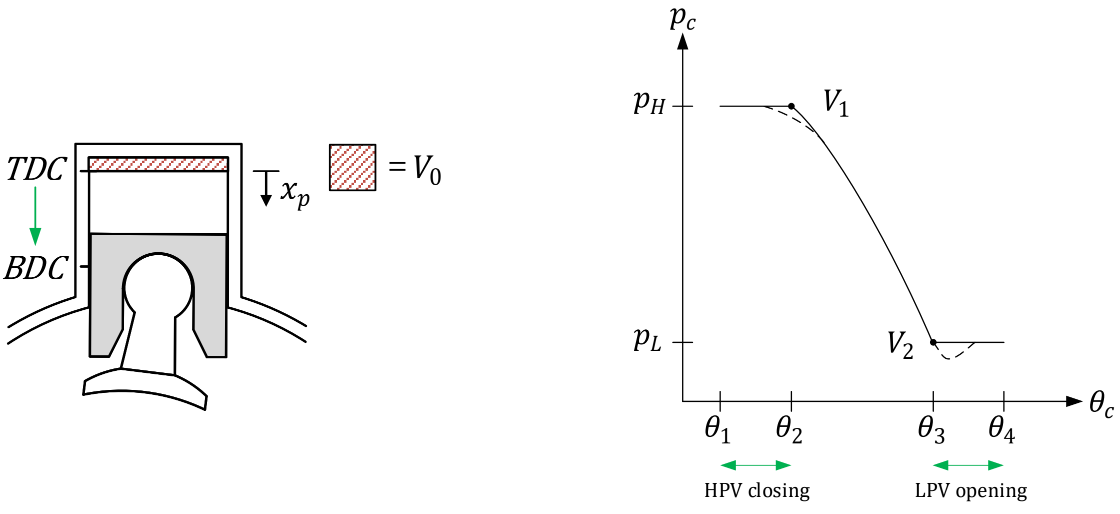
This section presents the general philosophy used for pressurization and de-pressurization of the cylinder chamber oil volume. For simplicity, the procedure is explained in motor mode rather than pump mode. To avoid large flow- and pressure peaks when switching the valves, internal chamber pressure must be brought to approximately the same level of either the high-pressure (p_H) or low-pressure (p_L) manifold. Done correctly, this lets the potential energy in the pressurised fluid do the pressurisation/de-pressurisation work. Correct de-compression and compression enables valve actuation at a low pressure difference over the valve, Δp_{valve}. The calculation of the angles is based on the method presented in [9].

The reader should note that:

- The size of the cylinder dead volume V_0 affects the distance travelled by the piston in order to pressurise/de-pressurise the fluid. In the model developed in this thesis, the dead volume is defined as $V_0 = V_d$
- The bulk modulus is kept constant in the presented calculations for simplicity, but is implemented as described in section 2.4 in the simulation model

2.6.1 De-pressurisation

Figure 2.12a shows a piston in its active period, moving from TDC to BDC. Hence, when the HPV closes and the active period is complete, the chamber volume V_c must be de-pressurized to p_L level, before opening the LPV.



(a) Downward piston movement

(b) Cylinder chamber de-pressurization

Figure 2.12: Decompression

Figure 2.12b plots the cylinder pressure, and shows how the timing of the valves must be executed in order to correctly decompress the fluid. The straight lines illustrate how the idealised cylinder pressure behaves based on the simplification introduced during the calculation that the chamber pressure p_c is constant during the opening and closing of the valves. The dashed lines show how the pressure in the cylinder actually behaves, where the pressure drops as the HPV reaches the closed position. Based on at which angle the sequence is initiated, θ_1, the proposed method can be used for calculating the opening angle of the LPV, θ_3.

θ_2 is calculated by the travelled angle during closing of the HPV:

$$\theta_2 = \theta_1 + t_s \cdot \dot{\theta}_m \tag{2.25}$$

Equation 2.7 is used to express the cylinder chamber volume, V_1, at angle θ_2:

$$V_1 = V_0 + \frac{V_d}{2}\left(1 - \cos\theta_2\right) \tag{2.26}$$

In the interval between θ_2 and θ_3, both valves are closed and the oil is de-pressurised. Thus, by use of the continuity equation, the following expression can be set up knowing that there is no flow in or out of the chamber:

$$\begin{aligned} \underbrace{Q_{in} - Q_{out}}_{=0} &= \dot{V} + \frac{V}{\beta} \cdot \dot{p}_c \\ -\dot{V} &= \frac{V}{\beta} \cdot \dot{p}_c \end{aligned} \tag{2.27}$$

Equation 2.27 is solved for the volume V_2, which is the cylinder volume at θ_3:

$$\begin{aligned}
-\frac{dV}{dt} &= \frac{V}{\beta}\,\frac{dp_c}{dt} \\
\int_{V_1}^{V_2} -\frac{1}{V}\,dV &= \int_{p_H}^{p_L} \frac{1}{\beta}\,dp_c \\
-\ln V_2 + \ln V_1 &= \frac{p_L - p_H}{\beta} \\
V_2 &= \exp\left(\ln V_1 - \frac{p_L - p_H}{\beta}\right) \qquad (2.28)
\end{aligned}$$

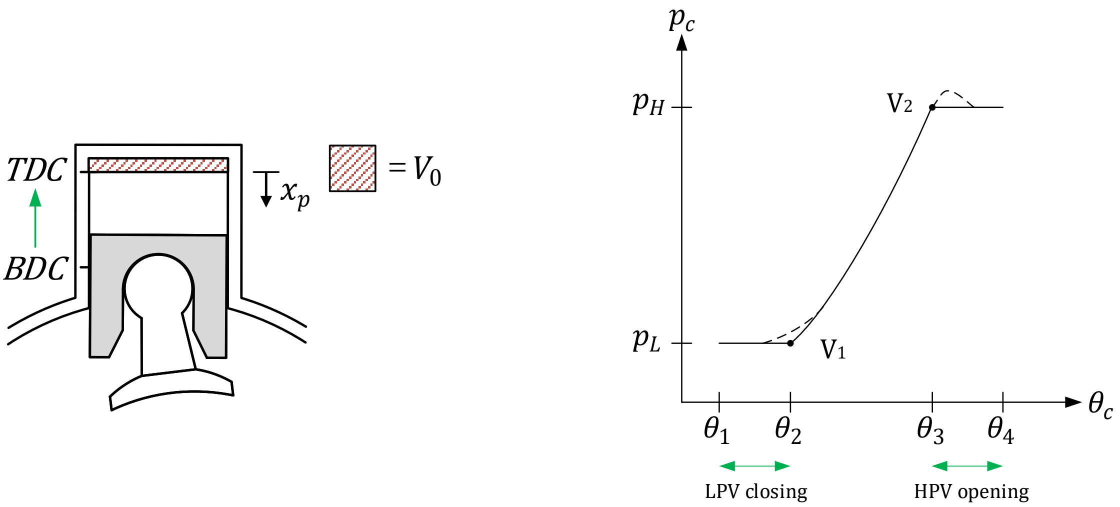
Then, by using the general equation for cylinder volume (equation 2.7), the critical LPV opening-angle θ_3 is calculated:

$$\theta_3 = \cos^{-1}\left(\frac{2}{V_d}\left(V_0 - V_2\right) + 1\right) \qquad (2.29)$$

As θ_3 is a function of the inverse cosine, the result may have to be subtracted from 2π depending on θ_1, as the inverse cosine function is defined in the interval $[0, \pi]$.

2.6.2 Pressurisation

Figure 2.13a shows a piston in its passive period moving from BDC to TDC. When the piston reaches TDC, the low-pressure oil volume must be pressurized from p_L to p_H level, to enable opening of the HPV at a low pressure difference, Δp_{valve}. Correct compression of the oil is crucial to prepare for the next working cycle.



(a) Upward piston movement

(b) Cylinder chamber pressurization

Figure 2.13: Compression

Figure 2.13b represents the chamber pressure during compression from p_L to p_H. The method for computing needed angles is similar to above. However, in motor mode, the sequence of pressurizing the oil must be completed right before TDC, at θ_3. Thus, opening of the HPV is initiated at θ_3, reaching completely open position at $\theta_4 = 2\pi$ (TDC). Calculations are therefore performed to produce the unknown LPV closing angle, θ_1.

The expression for θ_3 is:

$$\theta_3 = \theta_4 - t_s \cdot \dot{\theta}_m = 2\pi - t_s \cdot \dot{\theta}_m \tag{2.30}$$

The cylinder volume V_2 at angle θ_3 is expressed:

$$V_2 = V_0 + \frac{V_d}{2}\left(1 - \cos\theta_3\right) \tag{2.31}$$

The continuity equation (equation 2.27) is again applied, with two differences. The pressure limits on the integral are flipped, and the equation is solved for V_1:

$$V_1 = \exp\left(\frac{p_H - p_L}{\beta} + \ln V_2\right) \tag{2.32}$$

The volume V_1 is also a function of the angle θ_2. The general volume equation (equation 2.7) is rearranged to solve for θ_2:

$$\theta_2 = 2\pi - \cos^{-1}\left(\frac{2}{V_d}\left(V_1 - V_0\right) + 1\right) \tag{2.33}$$

The result from the inverse cosine function is substracted from 2π as the desired angle exists in $[\pi, 2\pi]$. Finally, the unknown LPV closing angle θ_1 is found:

$$\theta_1 = \theta_2 - t_s \cdot \dot{\theta}_m \tag{2.34}$$

3 | Control Challenge Analysis

This chapter presents the general secondary control strategies of DDMs, and discusses advantages and draw-backs when implementing these to the considered low-speed high-torque application. The possible load scenarios of the DDM are aslo presented. These demand the development of valve actuation strategies for different operation modes. Finally, a suggested control strategy is presented that incorporates the demands from the analysis.

Secondary control is a widely used term, and must therefore be clarified. This definition of secondary control is taken from Bosch Rexroth:

> Secondary control drives are operated in high-pressure networks. They consist of at least one hydraulic pump (the primary unit), a hydraulic accumulator, and a using unit (the secondary unit). The purpose of the hydraulic accumulator is to store the energy fed to the hydraulic network by the secondary unit whenever it is reversed, e.g. in the pumping mode. When operating under load, this accumulated energy is used to compensate for consumption peaks. The charge status of the hydraulic accumulator – in conjunction with the pressure-controlled primary unit and the operational status of the secondary unit – determines system pressure. [10]

For this particular project, it follows that all control strategy is applied directly to the DDM. It should also be noted that the HPU is not taken into consideration, thus the high-pressure (p_H) and low-pressure (p_L) sides are modelled as constant. Figure 3.1 shows the setup used for this project, with the HPU acting as the primary unit, and the DDM as the secondary unit:

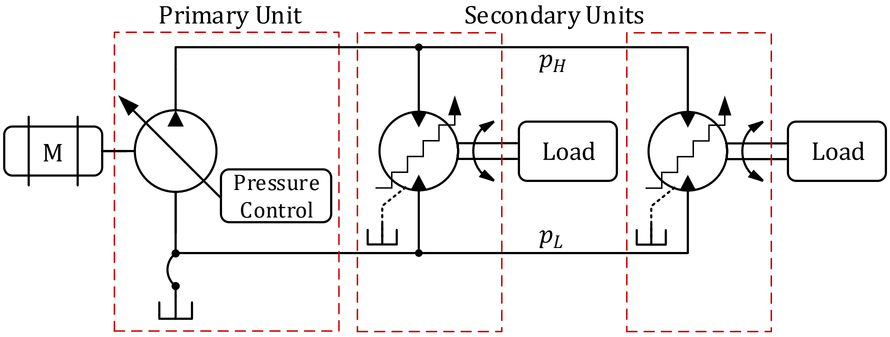


Figure 3.1: Schematic overview of hydraulic system

3.1 Displacement Strategies

This section discusses the two main displacement strategies that can be applied to a digital displacement motor. As mentioned in section 1.1.2, effective motor displacement can be changed by altering the state of a given set of the cylinders. An active state means that the cylinder is pressurised, thus set to produce torque. In an idle state, the cylinder is "de-activated" by closing the HPV and opening the LPV, allowing the piston to passively follow the movement of the rest of the motor while displacing low pressure fluid. This strategy is called full stroke displacement, since an active cylinder always utilizes the full stroke of the piston. The cylinder delivers torque in the active period, here defined from TDC to BDC. The piston then travels from BDC back to TDC, defined as the passive period, thus completing the full cycle. Each time a cycle is completed, the state of a cylinder can be switched to either active or idle.

The other displacement strategy is called partial stroke displacement. The fundamental distinction lies in the fraction of utilized piston stroke, where partial stroke strategy uses a displacement parameter describing stroke utilization, here called α. Hence, when using this displacement strategy, the active cylinder period is defined from TDC to α. This allows for better displacement resolution and controllability, but relies on pressurising all cylinders, in contrast to the full stroke strategy. This is further discussed in the following sections.

It is desirable to investigate the use of a DDM for a low-speed high-torque application, as stated in the problem statement (section 1.2). This practically means that for a DDM connected to a winch system, which operates at low speeds and requires high torques, the ratio between the motor shaft and the drum is 1:1, i.e. no gearbox. Consequently, the displacement strategies analysed in this section are evaluated at low rotational speeds to ensure their applicability to the considered load cases.

3.1.1 Full Stroke Displacement

Full stroke displacement strategy (FSS) uses a combination of active and idled cylinders in order to change the effective motor displacement, and thereby also the produced torque. If the state of a cylinder is changed from active to idle, the effective displacement of the machine is lowered. If the state of a cylinder is switched from idle to active, effective displacement is increased.

As mentioned, activated cylinders produce torque during the full piston stroke. An idled cylinder is simply connected to the low-pressure line during motor operation, and does not contribute to the overall motor torque. This is done by opening the LPV, allowing the piston to displace low-pressure oil as the active cylinders rotate the motor shaft. Thus, the low-pressure oil displaced by the idled cylinder is not a part of the motor's effective displacement. As mentioned, idled cylinders operate at low pressure, significantly reducing leakage losses over the pistons, compared to pressurised cylinders. This is in great contrast to the axial piston machines mentioned in section 1.1.1, where the cylinders are pressurised regardless of selected displacement.

Since the state of a cylinder can only be changed when a full cycle is completed, opening of the valves should be timed in conjunction with the piston, to open when the pressure difference across the valve is minimized. This gives the possibility of passive opening of the valves. Passive opening means that the pressure over the valve controls the valve plunger. For example, the HPV is passively opened when the cylinder chamber pressure, p_c, exceeds that of the high-pressure line, p_H. This can be seen in figure 3.2, where the HPV will move from closed to open position when $p_c > p_H$. Also, the LPV is passively opened when the chamber pressure drops below the level of the low-pressure line, p_L. Passive actuation of the valves contributes to a high system efficiency, since little to no electric power is used to switch the valve positions.

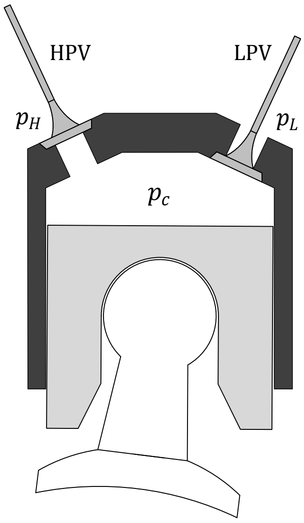


Figure 3.2: Valve orientation enabling passive opening

Figure 3.3 presents the general behaviour of one cylinder when FSS is applied to a digital displacement motor. The HPV and LPV positions are shown in dashed lines, x_H and x_L, where a value of one indicates fully open. Similarly, chamber pressure p_c and piston position x_p values are normalized, with maximum value equal to one. The plotted cylinder is initially in its active period, with the piston starting at TDC. Here, only the HPV is open, resulting in high chamber pressure and downwards piston movement. Hence, the cylinder produces torque. Near BDC, the HPV closes, resulting in decompression of the oil down to p_L level. Since $p_c \approx p_L$ at this point, the LPV is opened at a low pressure difference Δp_{valve}.

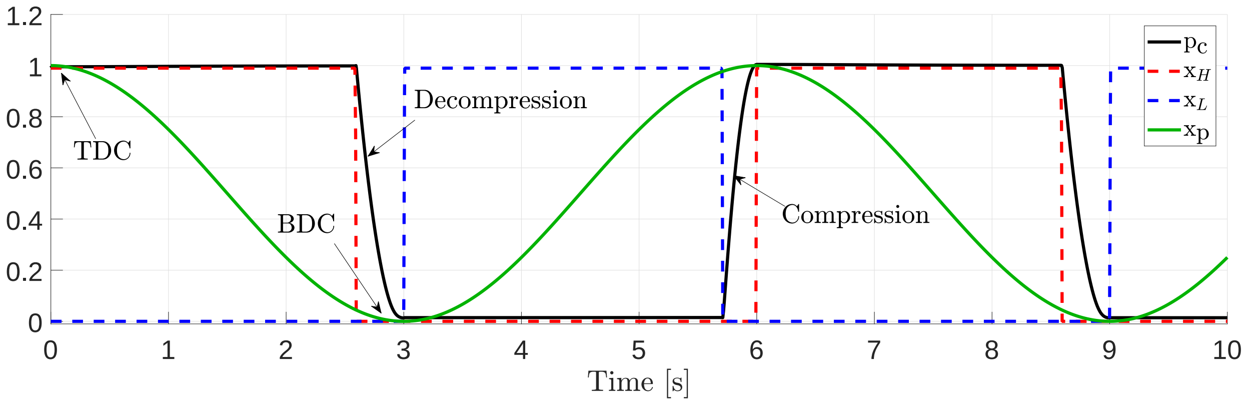
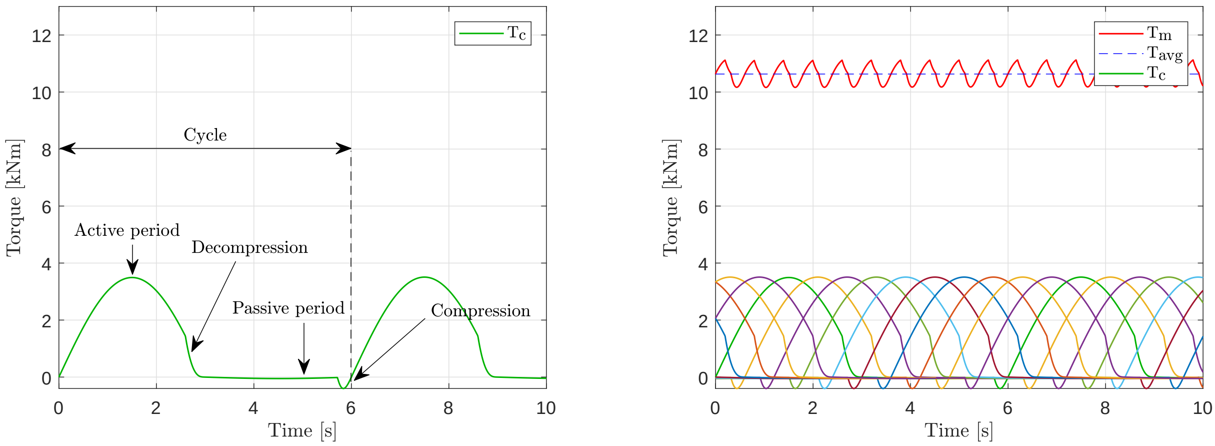


Figure 3.3: General full stroke displacement strategy

Then, the de-pressurised oil is disposed of through the LPV as the cylinder enter its passive period, and the piston moves towards TDC. In order to prepare for the next cycle, the low-pressure fluid must be compressed to p_H level near TDC. Compression of the oil is done by closing the LPV slightly before TDC. The HPV is opened directly after the compression sequence, since $p_c \approx p_H$ at this point, and the cycle is complete.

Figure 3.4 shows FSS used on a motor with ten cylinders operating at $10rpm$, where the torque produced by one cylinder, T_c, is shown in figure 3.4a. As can be seen, the cycle consists of the active period, decompression, passive period and compression. The cylinder produces torque in the active period, whereas the torque contribution is practically zero in the passive period. As can be seen, the decompression sequence limits the produced torque in the end of the active period. During the compression sequence, a small negative torque contribution is observed, due to the energy required to pressurise the oil. Torque contribution from all cylinders forms the resultant and average motor torque, T_m and T_{avg}, presented in figure 3.4b. As can be observed, the produced torque is relatively smooth since all cylinders are active. Smoothness of the resultant torque is dependent on overlap, i.e. number of cylinders.



(a) Single cylinder torque contribution

(b) Resultant and average torque

Figure 3.4: Ten-cylinder motor using full stroke displacement, operating at $10rpm$

When using FSS, motor displacement is changed by activating a given set of cylinders. Figure 3.5 shows the torque from the same ten-cylinder motor as presented above (figure 3.4). However, nine out of ten cylinders are set to active in this case. As can be seen, there is a large variation in resultant torque over a relatively large time period. At higher rotational speeds, this period might just act as a torque ripple, lowering the average torque, T_{avg}, seen by the motor. However, this example demonstrates problematic variations at lower speeds.

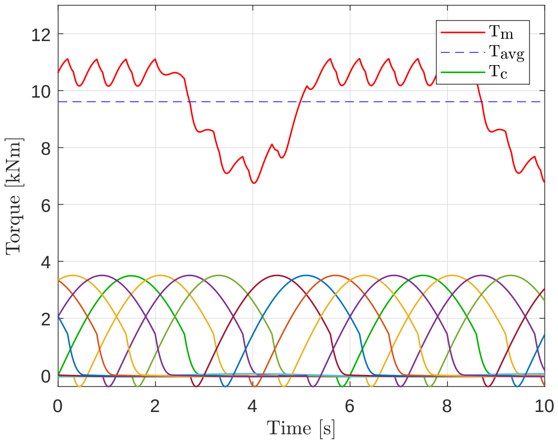
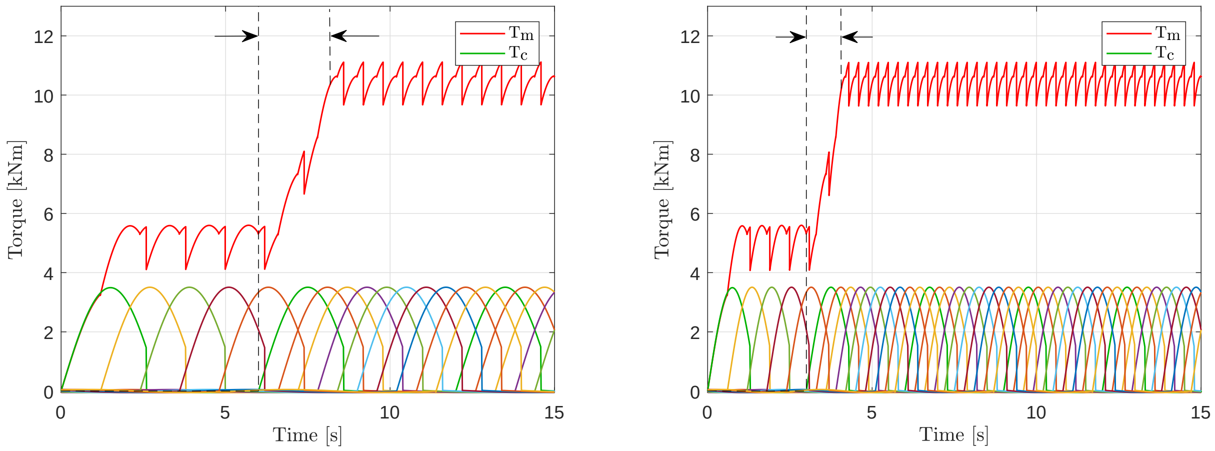


Figure 3.5: Nine out of ten cylinders active, $10rpm$

Further, an important property is how fast a motor can respond to a change in desired torque level, called torque response. Therefore, torque response of a digital motor operating with FSS is evaluated, in order to determine for which applications the strategy is suitable. Step-up torque response is referred to as how rapidly the motor can respond to a command of increased displacement, consequently increasing the produced motor torque. Similarly, step-down torque response is referred to as how fast the motor responds to a command of decreased displacement, which reduces the motor torque.

Figure 3.6a shows the ten cylinder motor given a step-up command from 50% displacement to 100% at $t = 6s$, while operating at $10rpm$. A displacement of 50% means that 5 cylinders are used. In this particular case, it takes approximately two seconds before all cylinders are activated and maximum torque is achieved. The torque response is dependent on rotational speed, since this determines how rapidly the cycles are completed. Figure 3.6b presents the same step in displacement at $t = 6s$, while the rotational speed is increased to $20rpm$. Compared to $10rpm$, the torque response is halved, to approximately one second. It should be noted that the torque response is not affected by magnitude of the displacement reference step, and that the step-down torque response will behave similarly.



(a) Torque response at 10rpm, approximately 2 sec delay

(b) Torque response at 20rpm, approximately 1 sec delay

Figure 3.6: Full stroke displacement step-up torque responses

As presented, FSS is challenged when applied to digital motors operating at low rotational speeds. For example in figure 3.6a, the response time is approximately two seconds. The response time can be as high as one half shaft revolution, depending on at which point in the cylinder cycle the command signal is given [11, p. 13]. Due to the mentioned point where a cylinder can change state (active or idle), system bandwidth is dependent on the rotational speed and piston frequencies, meaning how often a piston reaches the decision point at TDC. Hence, the representing control challenge must be counteracted by a high number of cylinders or possibly a cam ring motor design.

3.1.2 Partial Stroke Displacement

As the name implies, partial stroke displacement strategy (PSS) utilizes a desired fraction of the piston stroke in order to change displacement. This gives increased displacement resolution, compared to the discrete behaviour characterized by FSS. The displacement is continuously regulated by adjusting a displacement parameter, here called α. Thus, the active period where a cylinder produces torque is defined from TDC to α. In other words, α represents a point where the active period is limited. When using maximum motor displacement, α is set to its maximum value, α_{max}. Then, the entire piston stroke is utilized, and the duration of the active cylinder period corresponds to that of FSS.

Figure 3.7 shows the general behaviour of PSS applied to a digital displacement motor. Similar to figure 3.3, the HPV and LPV valve positions are shown in dashed lines, x_H and x_L, where a value of one indicates fully open. Also, chamber pressure p_c and piston position x_p values are normalized, with maximum value equal to one. Here, the displacement parameter α is set to 70% of the maximum value α_{max}, giving partial displacement. Thus, the active period is limited to a fraction of the piston stroke, done by closing the HPV and initiating the decompression sequence. As can be seen, the decompression sequence is more rapidly executed, compared to when utilizing the full piston stroke (figure 3.3). This is due to higher piston velocity at the point where the HPV is closed, and therefore a higher rate of change in cylinder volume, $\dot{V}_c$. After decompression, the cylinder chamber pressure p_c is close to the level in the low pressure line p_L, and the LPV is opened. Low-pressure oil is drawn into the chamber in the remaining piston travel towards BDC. The subsequent oil disposal and compression sequence is analogous to FSS.

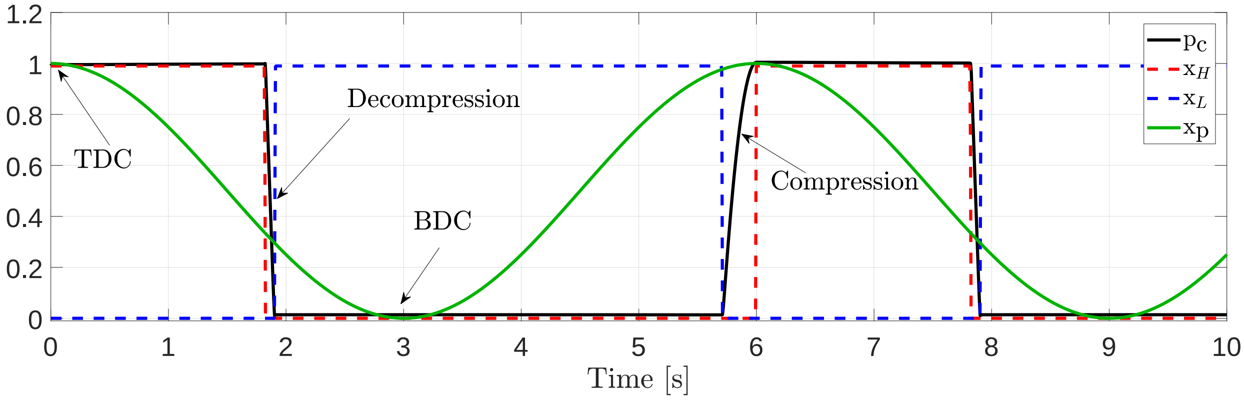
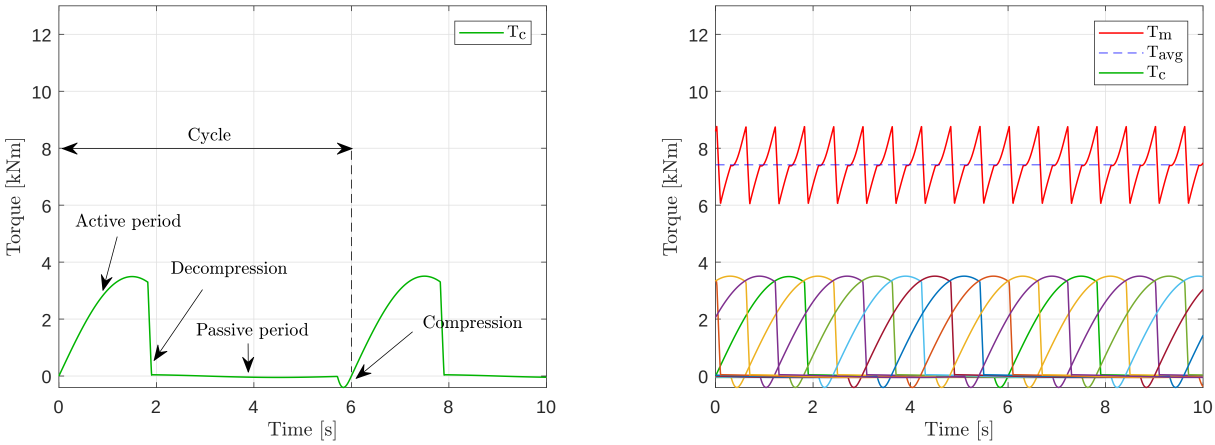


Figure 3.7: General partial displacement strategy

Figure 3.8 shows PSS applied to a ten-cylinder motor operating at $10rpm$, with $\alpha = 70\% \cdot \alpha_{max}$. As can be seen in figure 3.8a, a full cycle consists of the active period, decompression, passive period and compression. The cylinder now produces torque in the active period ranging from TDC to α, giving the characteristic shape of the torque plot, observed at $t \approx 1.9s$. Next, a short decompression sequence takes place, followed by the passive period where the torque contribution is zero. During the compression sequence, a minor negative torque contribution is observed, due to the energy required to raise the oil pressure to p_H level.

Figure 3.8b presents the resultant motor torque T_m and average motor torque, T_{avg}. The resultant torque behaves differently for PSS than for FSS when the displacement is altered, as can be seen when comparing figure 3.8b with figure 3.5. There is no drop in torque, but the ripples have a higher amplitude than those observed for the torque plots with FSS. This is due to the chosen α that de-activates the cylinders close to their maximum torque contribution, as seen in figure 3.8a.

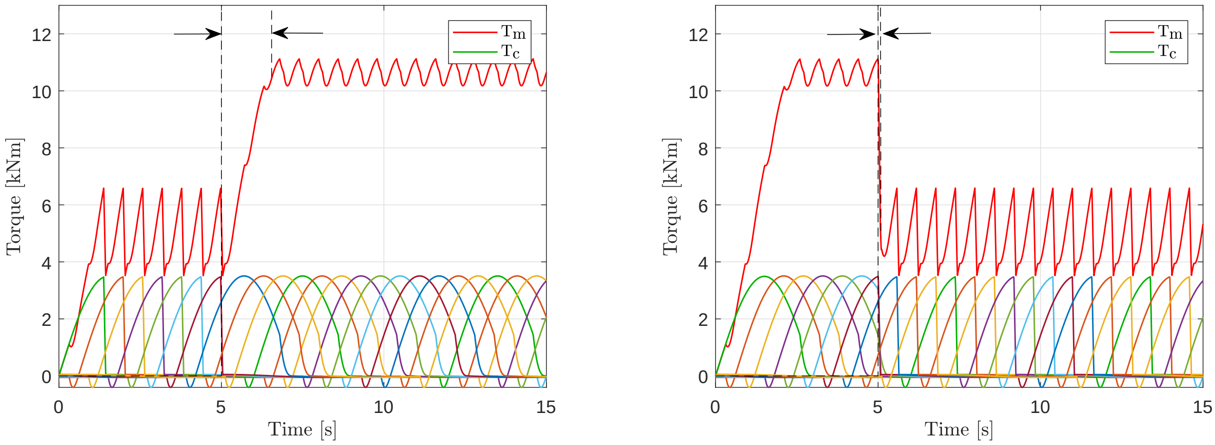


(a) Single cylinder torque contribution

(b) All cylinders and resultant torque

Figure 3.8: Ten-cylinder motor using partial stroke displacement, at 10rpm

To determine for which applications PSS is suitable, motor torque responses are evaluated. When the system is given a step in α, the motor will change displacement after a given time delay, and thereby also the produced torque. Figure 3.9 shows a digital motor operating at 10rpm, while using PSS. Figure 3.9a presents the step-up torque response of the motor. Initially, α is set to 50% of its maximum value, giving partial displacement. At $t = 5s$, the system is given a step-up to α_{max}, where the full piston stroke is utilized. As can be seen, the torque response is approximately 1.5 seconds. When compared to the motor using FSS under similar conditions (figure 3.6a), the PSS torque response is 0.5s faster. Figure 3.9b shows the PSS step-down torque response, from α_{max} to 50% of α_{max}, showing an almost instantaneous reaction. This is because α denotes the end of the active period, thus any cylinder outside of the newly reduced active period is de-activated immediately. In should be noted that when using PSS, the step-up torque response is dependent on rotational speed, in addition to size of the applied reference signal.



(a) PSS step-up torque response - 1.5sec

(b) PSS step-down torque response - 0.25sec

Figure 3.9: PSS torque responses - 10rpm

Figure 3.10 presents how the valves are actuated and the behaviour of the related cylinder pressure, during the step-up in displacement presented in figure 3.9a. As can be seen, the first cycle uses part of the piston stroke. Then, the displacement is increased by utilizing the full piston stroke in the next cycle.

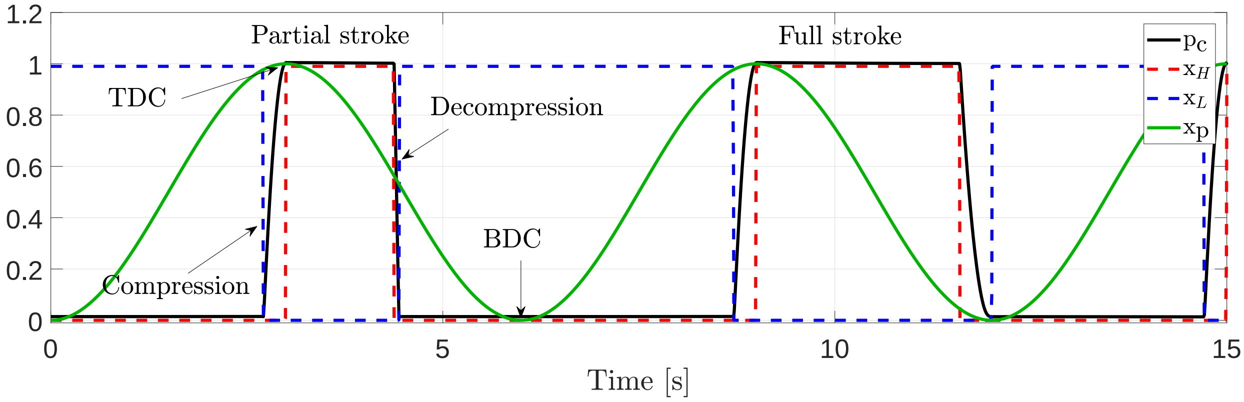


Figure 3.10: Increase in motor displacement

When evaluating the presented torque responses using partial displacement, it is clear that the most challenged case is the step-up sequence (figure 3.9a). It is therefore further investigated whether the time delay of 1.5 seconds can be improved. To further improve the step-up torque response, cylinders that had finished producing torque up to a given α can be reactivated if a higher value of α follows. For example, the cylinder in figure 3.10 was set to produce a torque up to 50% of α_{max}. At $t = 5s$, α is increased to α_{max}, which is prior to the piston reaching BDC. Thus, if the cylinder is allowed to be reactivated when a higher value of α is set, it is still possible for this particular cylinder to produce torque in the remaining piston stroke. Thus, the overall motor torque response is improved.

If the cylinder in figure 3.10 is allowed to reactivate when α is increased, the behaviour is changed to the one found in figure 3.11. As can be seen, the active period for the cylinder is originally set to finish at 50% of α_{max}. At $t = 5s$, α is increased to its highest value, prior to the piston reaching BDC. Thus, the remaining fraction of the piston stroke is utilized by reactivating the cylinder. It is important to notice that the reactivation sequence does not enable proper pressurisation of the chamber, meaning that the valves are actuated at high pressure difference.

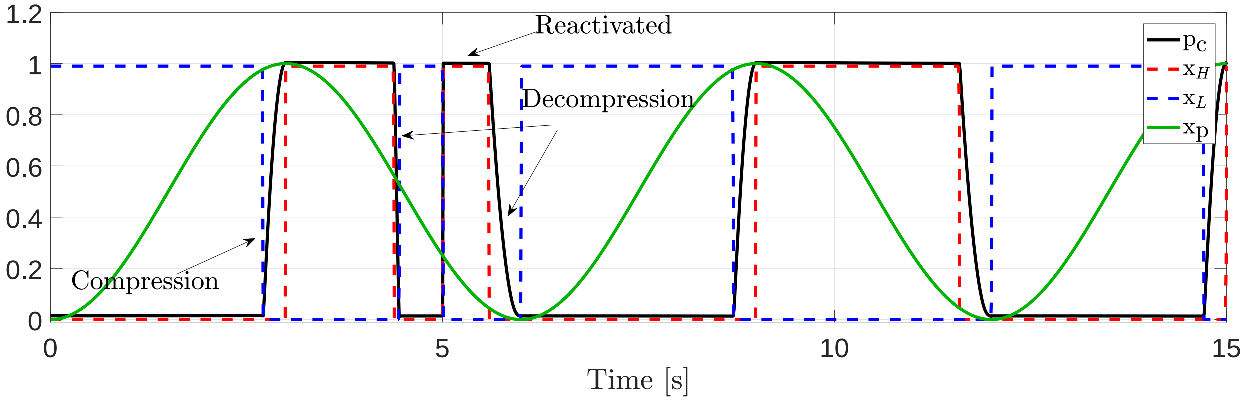
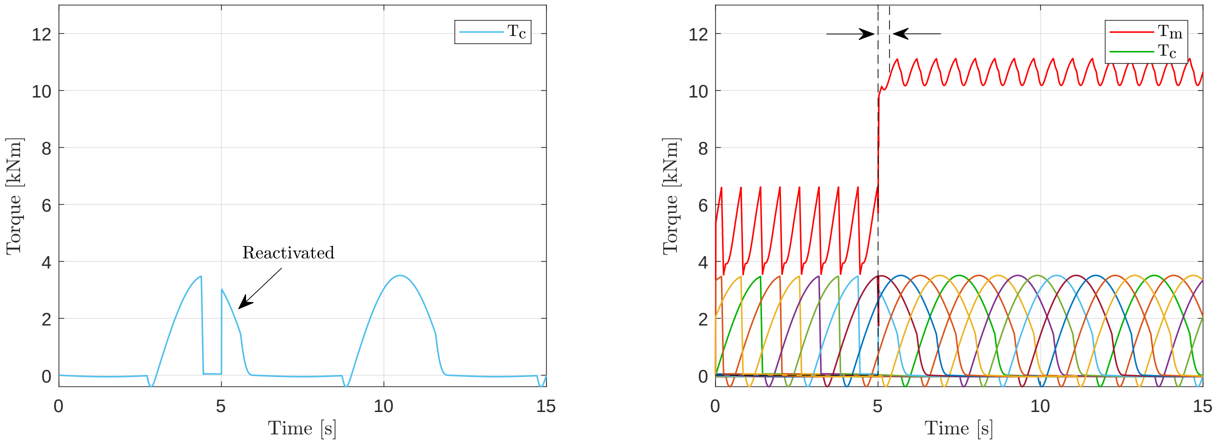


Figure 3.11: Re-activating the cylinder

By allowing the cylinder to reactivate, the torque contribution from the cylinder becomes as in figure 3.12a, where the characteristic behaviour is clearly visible. When the respective cylinders are allowed to reactivate, the motor torque response becomes as in figure 3.12b, showing a step response of approximately 0.35s. The step-response is now essentially determined by the time it takes for the pressure to build up the cylinder chamber. Compared to the original step response in figure 3.9a, the torque response is vastly improved, by approximately 1.15s.



(a) Re-activated cylinder

(b) Improved step-up response

Figure 3.12: Reactivated cylinder - improved torque response

As presented, partial stroke displacement strategy gives faster system response when compared to a system using full stroke displacement strategy under similar conditions. Further developing the partial stroke strategy by allowing cylinders to be reactivated when possible resulted in a significant improvement of the step-up torque response.

However, when the cylinders are reactivated, they can not be pressurized to facilitate valve switching at low pressure difference, Δp_{valve}. A significant pressure difference over the valve during switching translates into a force that opposes the opening or closing motion. Presumably, pilot-stage seat valves are needed to enable switching at these conditions. Also, flow peaks will occur when switching under these conditions, due to oil rushing in the direction of lowest pressure.

A mentionable disadvantage of using PSS is that the cylinders will be pressurised each cycle, resulting in leakage losses over the pistons and lowering system efficiency. However, due to the considered winch system in this project, the motor torque response is recognized as one of the most important properties. This is due to the direct connection between the motor and winch drum, meaning lower rotational speeds.

3.2 Control System Requirements

This section presents an analysis of the considered application for the DDM, and outlines the main requirements needed to deal with the most important aspects of the control problem.

Since the considered winch system is connected directly to the motor shaft, the DDM will operate at lower rotational speeds. Therefore, full stroke displacement strategy is expected to give a slow torque response. Also, the problem statement (section 1.2) states the importance of steady handling and the ability to follow a typical active heave compensation movement. In order to obtain desired response and accuracy for the winch, partial stroke displacement strategy as described in section 3.1.2 with the ability to re-activate cylinders is selected for further use in this project.

Depending on the load conditions, the DDM can in practice operate as a motor or a pump. For example, if the motor is connected to a winch that is set to lower a payload, the resulting load torque works in the same direction as the motor speed. In order to perform a controlled lowering, the motor must act as a pump to counteract the load torque. Contrarily, if the winch is set to lower the hook with no load, the weight of the spun out wire may not be enough to overcome the friction in the system. In this case, the DDM must work as a motor to lower the unloaded hook. Figure 3.13 shows an operational chart that structures the load cases in four quadrants:

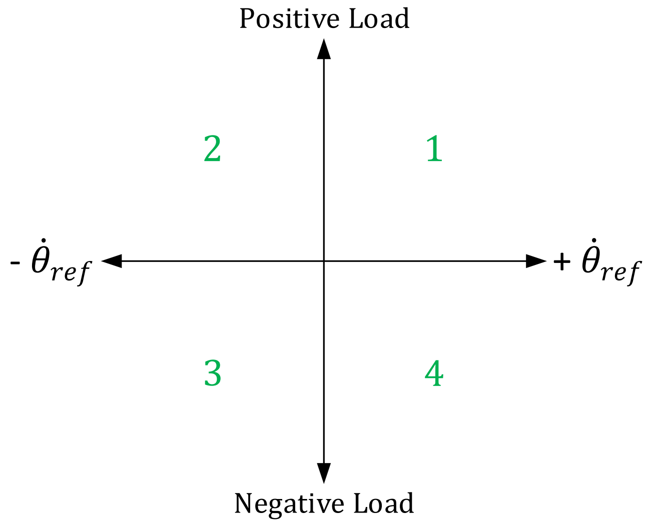


Figure 3.13: Four Quadrant Operation

Positive rotation is defined as counter-clockwise. A load that is acting in the opposite direction of the motor rotational speed is defined as a positive load. In this case, the DDM must apply a torque in order to actuate the load in the desired direction, hence it supplies energy and is therefore a motor. Refering to figure 3.13, this corresponds to the two upper quadrants. A load that is acting in the same direction of the motor rotational speed is defined as a negative load. The applied torque from the DDM serves to counteract the load torque. The system receives energy from the load and the DDM therefore works as a pump. Referring to figure 3.13, this scenario corresponds to the two lower quadrants.

The control system must therefore have a defined set of operation strategies for each quadrant in order to cover all possible scenarios. If the load and/or the velocity reference changes direction, the control system should be able to detect the change and act accordingly, i.e. determine the operating quadrant and switch between the defined operation strategies.

3.3 Suggested Control Strategy

The suggested control system for the DDM consists of three main interdependent subsystems that regulates the system behaviour at different levels. A traditional PID-controller based on velocity feedback to determine the displacement parameter α, a mode selector that determines what quadrant the DDM should operate in, and a set of algorithms for each quadrant that determines the timing of the closing/opening of each valve according to the given α. A feed-forward element is added to the PID-controller. The basic controller structure showing is shown in figure 3.14:

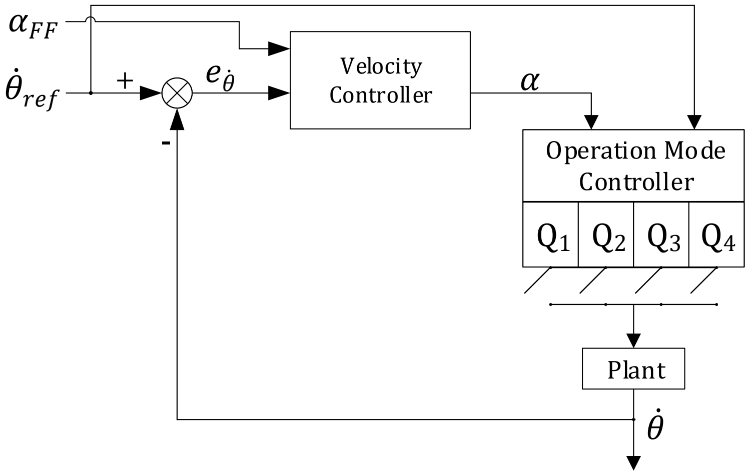


Figure 3.14: Block diagram of control system

As focus in this thesis is directed at the low level control structure, a simple PI-controller is favoured over a more complex solution for the reference tracking. This probably comes at the cost of some reduced reference tracking accuracy, which at reasonable levels is regarded as an acceptable trade-off for this project.

The displacement parameter α is an angle that determines at what point the cylinders are deactivated or activated, for motor and pump mode respectively. As the velocity error can be negative, the output from the velocity controller can also be negative. A negative α simply implies that the torque from the motor is negative (applied in CW direction).

With this information, the control system knows what direction it wants to apply torque, and the direction of the desired velocity. Based on the quadrant figure 3.13, the operation mode of the DDM and its corresponding control scheme can be determined. The different operation modes are presented before the mode operation controller. This is done to help the reader understand what defines the different modes, as the mode operation controller makes decisions based on criteria that characterises the different quadrants. It should be noted that in this thesis, the machine is subsequently termed as a motor, even if it in some cases in practice works as a pump.

3.3.1 Modes of Operation

This section explains in detail how the different modes of operations and their respective valve timing strategies work. The valve timing strategies for each mode are illustrated with valve diagrams that depicts the status of the valves and the pressure in the cylinder chamber. In the diagrams, these properties are related to the cylinder piston position, represented by the cylinder angle θ_c, as it runs through the cycle of the mode.

First Quadrant: Counter-Clockwise Motoring

This mode is the standard CCW motor operating mode where the load torque is working in the opposite direction than the motor velocity. Figure 3.15 illustrates this scenario:

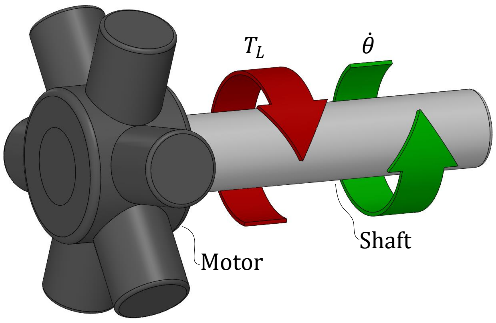


Figure 3.15: First quadrant, CCW motoring

The valve timing diagram in figure 3.16 presents the whole cycle for a cylinder during CCW motor mode. The figure describes how the cylinder is pressurised during the cycle of the cylinder angle θ_c, starting at TDC. The cylinder angle θ_c runs from 0 to 2π. The inner circle represents the pressure of the fluid in the chamber, where red is high pressure and blue is low pressure. The parameter α is shown on the left-hand half of the diagram, and denotes the end of the active period of the cylinder. The outer circle shows the state of the HPV (dark grey) and the LPV (light grey). The slopes at the start/end of their open period indicate the angles needed to close/open the valves. The white area in between the grey areas on the outer circle implies that both valves are closed, i.e. the fluid is either under compression or de-compression.

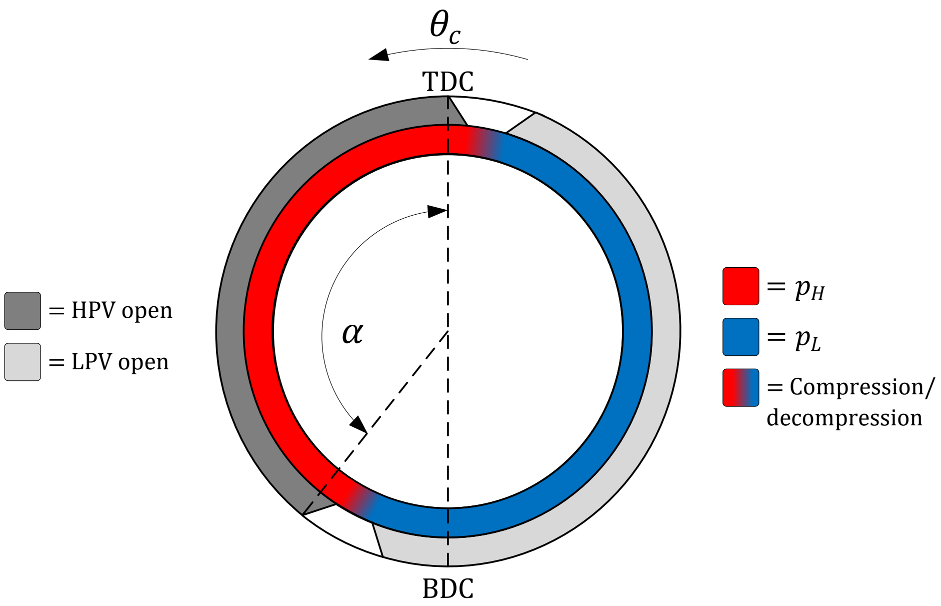


Figure 3.16: Valve diagram for 1st quadrant, where $\dot{\theta}_m > 0$

The cycle starts at TDC where $\theta_c = 0$, the HPV is fully opened and the cylinder chamber is pressurized. The HPV is open during the active period when θ_c is between TDC and α. When the cylinder angle θ_c reaches α, the HPV closes in order to decompress the fluid. The LPV opens when the cylinder pressure has dropped to the level of the low pressure side, as shown in the general de-pressurisation sequence from section 2.6. It remains open until the compression sequence takes place right before TDC.

In a scenario where the control parameter α decreases to some point where $\theta_c > \alpha$ when a cylinder is in the active period, the cylinder initiates the de-pressurisation sequence immediately and the control system calculates the necessary angles to time the de-pressurisation sequence according to the procedure described in section 2.6.

In a scenario where the cylinder is in its passive period before the piston reaches BDC, it can still be re-activated by the control system if the control parameter α increases in value to some point where $\theta_c < \alpha$, as mentioned in section 3.1.2. In that case, the cylinder piston will not be able to pressurise the fluid since the piston is travelling downwards. The HPV is therefore opened at the same time as the LPV is closed in order to pressurise the cylinder chamber as fast as possible, resulting in some flow peaks over the two valves.

Second Quadrant: Clockwise Motoring

This mode is the CW motor operating mode. As for the first quadrant, the load torque is working in the opposite direction than the motor velocity. Figure 3.17 illustrates this scenario:

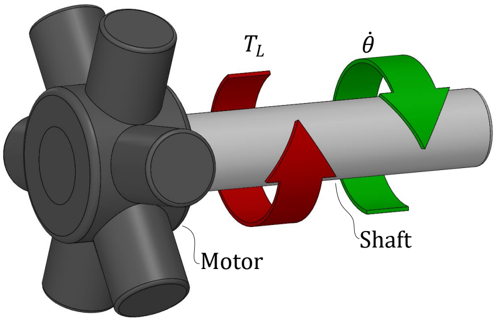


Figure 3.17: Second quadrant, CW motoring

Theoretically, the load on a motor could be applied in any direction. For a winch application, this implies that the load is acting in opposite direction of the gravitational force, which is not a realistic scenario. However, a light load/no load on the hook could produce a similar scenario as there might be friction in the system that the light load is unable to overcome. Thus, it can be necessary to assist the downward motion during lowering. The DDM must assist in accelerating high levels of rotational inertias within the winch system, that otherwise had to be sat in motion by the light payload alone. Failing to do so might result in deviation from a given trajectory during lowering of the payload.

As both the first and second quadrants operate as motors, they share the same valve timing structure. The DDM must work in the same way as in the first quadrant, only that the rotational speed is reversed. To apply a negative torque to drive the motor in the clockwise direction, the parameter α is defined between 2π and π, as shown in figure 3.18. Thus, the active period of the cylinder is on the right-hand side of the valve diagram.

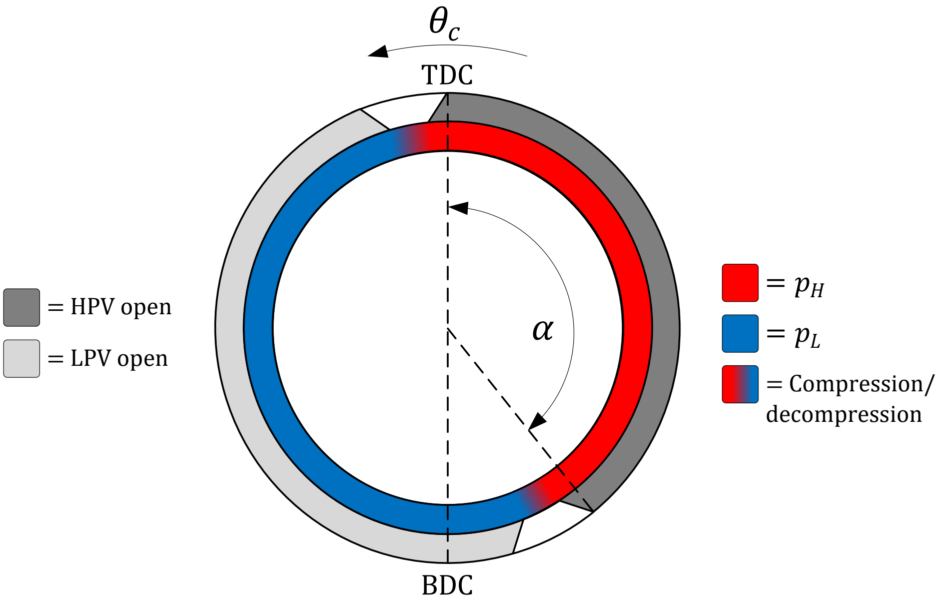


Figure 3.18: Valve diagram for 2nd quadrant, where $\dot{\theta}_m < 0$

The cycle for this operation is analogous to that of the first quadrant, only mirrored over the TDC-BDC axis as shown in figure 3.18.

Changes in the control parameter α affects the cylinder in the same manner as in the first quadrant. This means that a decrease in value initiates the de-pressurisation sequence of an active cylinder, and an increase in value immediately re-activates available cylinders, leading to the same scenario as described for the first quadrant, with resulting flow peaks.

Third Quadrant: Clockwise Pumping

This mode is the standard CW pump operating mode where the load torque acts in the same direction as the motor velocity, shown in figure 3.19. During lowering of the payload, the DDM must work as a pump to secure a controllable operation. Together with the first quadrant, this operation mode is expected to be dominant for a typical winch application where the load is heavy enough to accelerate itself downwards due to gravity.

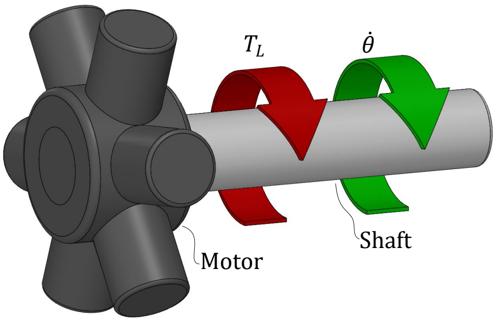


Figure 3.19: Third quadrant, CW pumping

As the load itself rotates the motor shaft, the control system finds a suitable α in order to lower the load according to a set motion reference. As depicted in figure 3.20, the active period of the CW pump mode is on the left-hand half of the diagram. In contrast to the motor mode, the parameter α now denotes the *start* of the active period, as the motor is rotating in the CW direction.

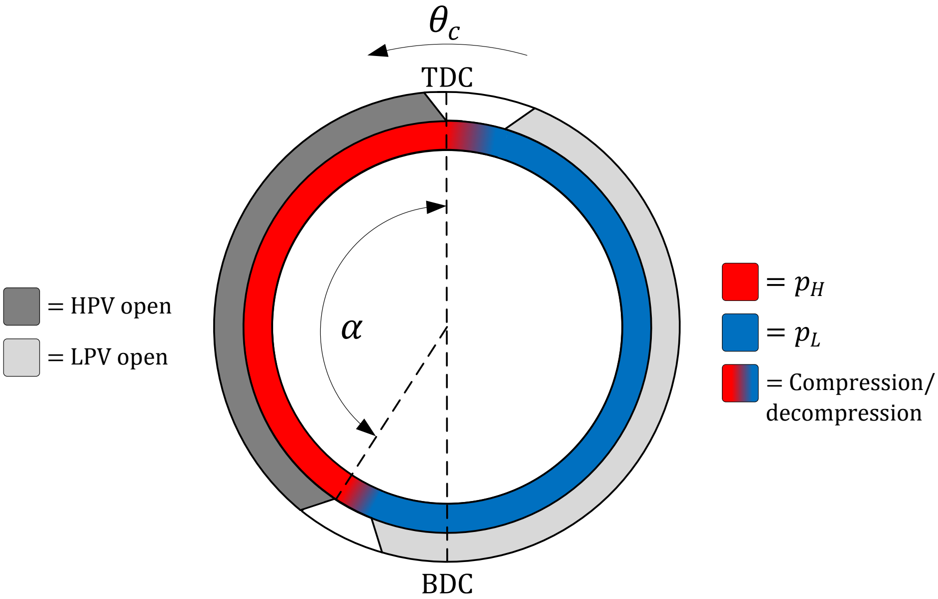


Figure 3.20: Valve diagram for 3rd quadrant, where $\dot{\theta}_m < 0$

The cycle starts at $\theta_c = \alpha$ with a fully pressurized chamber and the HPV opens. The payload works against the DDM cylinders, and creates a slightly higher pressure in the chamber than in the high pressure manifold in order to move fluid back over the HPV. The HPV then remains open until the de-pressurisation sequence. When the piston reaches TDC, it will start its downward motion. Thus the closing sequence of the HPV is initiated right before TDC in order to prevent the cylinder from producing torque in the opposite direction. As the piston moves past TDC, the fluid is decompressed and the LPV is opened at a low pressure difference.

The LPV remains open until the closing angle of the LPV associated to the α set by the control system is reached. The cylinder is then pressurised by the payload before the HPV again opens at a low pressure difference.

In pump mode, the cylinder behaviour differs from how it reacts in motor mode when a significant change in the control parameter α is observed. In a scenario where the control parameter α increases to some value where $\theta_c < \alpha$, the LPV immediately closes and the control system calculates the opening angle of the HPV needed to open against a low pressure difference.

If the control parameter α decreases to some value where $\theta_c > \alpha$, an active cylinder in pump mode is de-activated before it reaches TDC. As the piston in the active period of a pump mode is travelling upwards, the de-pressurisation of the cylinder chamber can not be performed in the controlled manner described in section 2.6. The LPV is therefore opened at the same time as the HPV is closed, resulting in similar flow peaks as for a re-activated cylinder in motor mode.

Fourth Quadrant: Counter-Clockwise Pumping

This mode is the CCW pump operating mode, where the load torque acts in the same direction as the motor velocity. Similarly to the load case describing operation in the second quadrant, a load acting in CCW direction over time is not a realistic situation for a winch. Figure 3.21 shows how the combination of load torque and motor velocity looks like for the fourth quadrant.

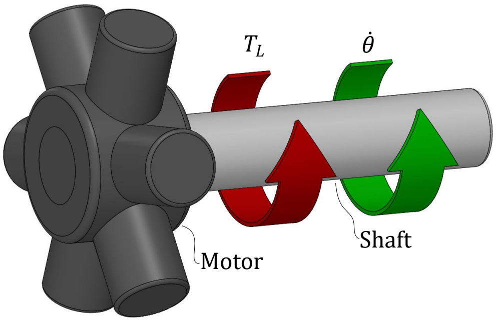


Figure 3.21: Fourth quadrant, CCW pumping

However, in a scenario where the payload somewhat overshoots a reference set by the controller in an upwards motion, the high inertia of the drum combined with a light load connected to the winch may lead to a significant reference error. This could be counteracted by switching the DDM over in CCW pump mode in order to absorb the energy from the rotating inertia. Figure 3.22 shows the valve diagram for this mode:

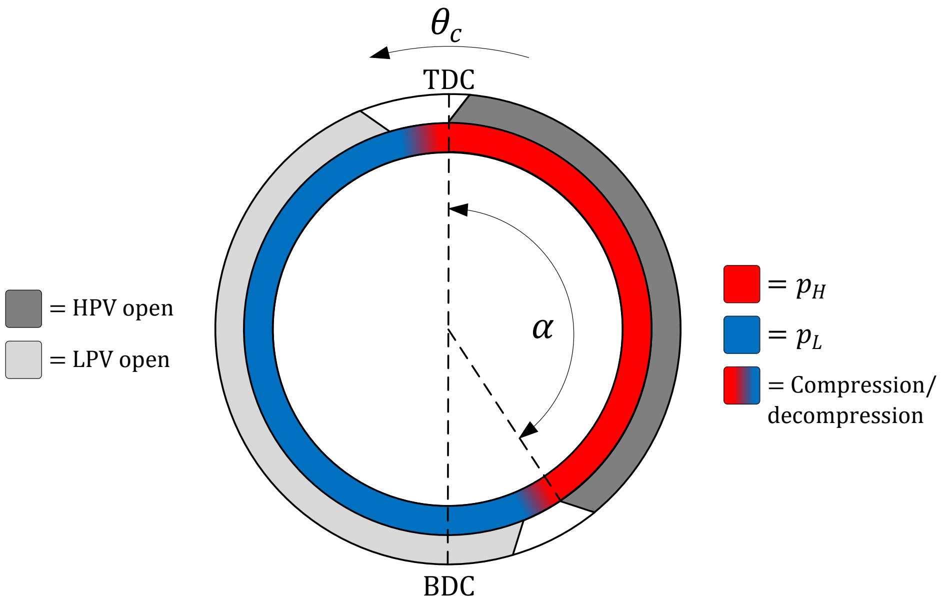


Figure 3.22: Valve diagram for 4th quadrant, where $\dot{\theta}_m > 0$

As both the third and fourth quadrants operate as pumps, they share the same valve timing structure. The DDM must work in the same way as in the third quadrant, only that the rotational speed is reversed. The cycle for this operation is analogous to that of the third quadrant, only mirrored over the TDC-BDC axis. To do this, the parameter α must be defined between 2π and π. Thus, the active period of the cylinder is on the right-hand side of the valve diagram.

Changes in the control parameter α affects the cylinder in the same manner as in the third quadrant. This means that a increase in value initiates the pressurisation sequence of a passive cylinder (where $\theta_c <$ BDC), and a decrease in value immediately de-activates an active cylinder leading to the same scenario as described for the third quadrant, with resulting flow peaks.

3.3.2 Operation Mode Controller

The operation mode controller determines what quadrant the motor should operate in. As mentioned in the beginning of section 3.3, the sign of the displacement parameter α determined by the control system indicates the direction of the motor torque. By assuming that the desired motor torque is applied in opposite direction of the load torque, the operating quadrant can be determined by evaluating the sign of α and the sign of the desired velocity, the velocity reference $\dot{\theta}_m$. The choice the controller makes is based on figure 3.13. Figure 3.23 shows a flow chart of the decision-making process of the controller. Tables 3.1 and 3.2 sums up the conditions for each operation mode.

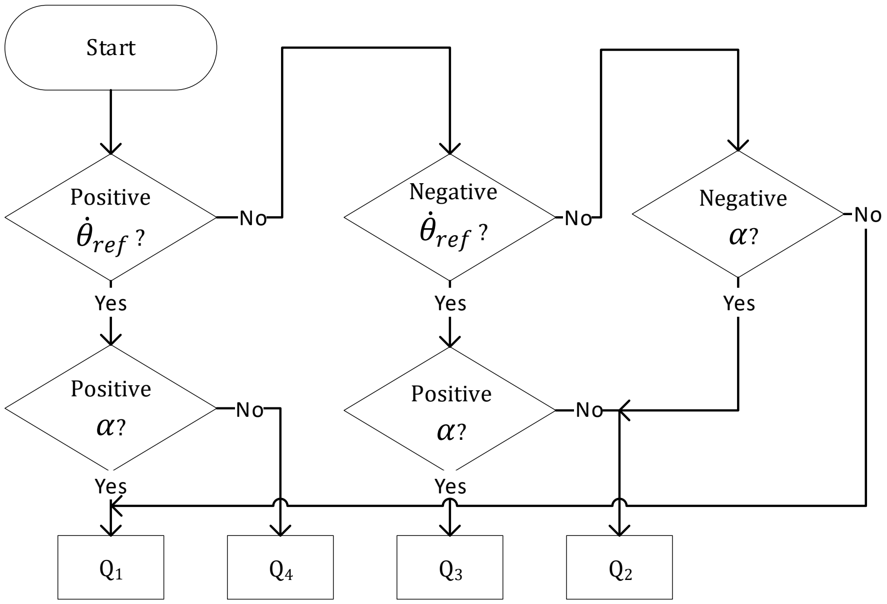


Figure 3.23: Flow chart operation mode controller

Q	$\boldsymbol{\dot{\theta}_{ref}}$	$\boldsymbol{\alpha}$
$\oplus^1$	+	+
${}^2\oplus$	-	-
${}_3\oplus$	-	+
$\oplus_4$	+	-

Table 3.1: Normal conditions

Q	$\boldsymbol{\dot{\theta}_{ref}}$	$\boldsymbol{\alpha}$
$\oplus^1$	0	+
${}^2\oplus$	0	-

Table 3.2: Special case

The first column in the tables contains the operation mode, and the second and the third column describes the direction of the velocity reference and the desired torque. The special case where $\dot{\theta}_{ref}$ is neither positive or negative refers to an angular velocity reference of 0 $\frac{rad}{s}$. In that case, the controller decides to work as a motor in either the 1st or the 2nd quadrant depending on α.

Some challenges are expected to occur when the motor switches between the different operation modes, especially if the operation mode is changed while a cylinder is in the middle of its compression/decompression sequence. The mode switching must therefore be simulated under specific operating conditions that forces the control system to switch mode, in order to identify the switching challenges and ensure an acceptable transition.

4 | Considered Winch System

To test the motor functionality on a relevant industrial application, a simplified off-shore winch model is set up. This chapter presents how the winch system is modelled, and analyses the considered load case scenario.

As this thesis mainly focuses on the development of the control system for the DDM, the modelling detail of the winch is limited to include the main properties that characterises the dynamic behaviour of the load. More realistic elements such as a knuckle-boom crane, sheaves, a travelling block and a passive drum only brings unnecessary mechanical complexity without great effect on the dynamical behaviour of the payload. Thus, the components required to form a simplified winch assembly is a winch drum, steel wire and payload. Figure 4.1 presents the considered winch configuration:

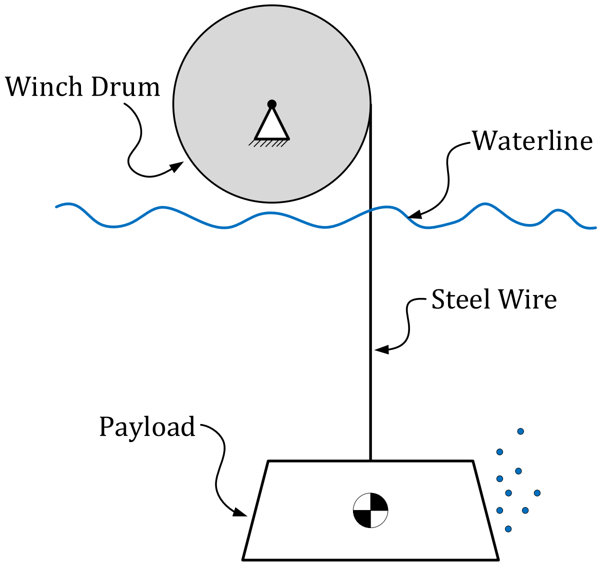


Figure 4.1: Winch Overview

The winch system parameters presented in table 4.1 are obtained from a regional industrial off-shore supplier. The winch is designed to handle a payload of 20 tonne at a maximum payload velocity of $\dot{z}_{pl,max} = 1.5\frac{m}{s}$, in addition to the roughly 20 tonne of wire rolled onto the drum.

Parameter	Dimension	Description
d_d	1632mm	Drum diameter
L_{wd}	3660m	Wire on drum
w_d	1260mm	Drum width
d_w	34mm	Wire diameter
ρ_w	5.66kg/m	Wire mass per meter
J_d	1111kg·m^2	Drum inertia

Table 4.1: Winch parameters

4.1 Steel Wire & Payload

The steel wire is modelled as a parallel spring-damper system that suspends a payload mass m_{pl}, as shown in figure 4.2. Also, the wire mass, m_w, is considered as lumped mass distributed above and below the region of the spring-damper:

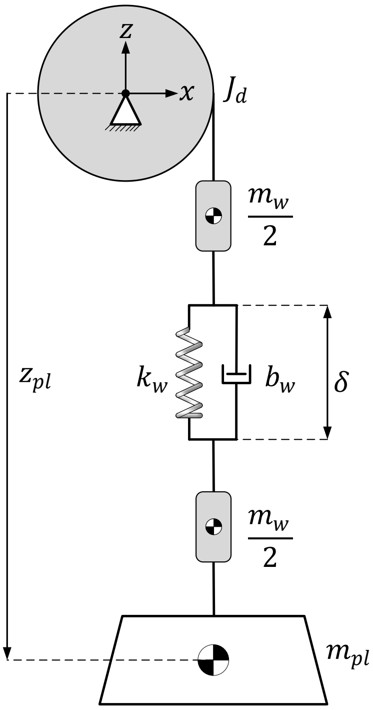


Figure 4.2: Steel wire model

As the steel wire is modelled as flexible, the wire force is a function of wire elongation and elongation rate, shown in equation 4.1.

$$F_w(\delta, \dot{\delta}) = \delta \cdot k_w + \dot{\delta} \cdot b_w \tag{4.1}$$

where

F_w	-	Wire force
δ	-	Wire elongation
$\dot{\delta}$	-	Wire elongation rate
k_w	-	Wire stiffness
b_w	-	Wire damping coefficient

It then follows that for $\delta < 0$, then $F_w = 0$ since the wire does not compress. Stiffness of the wire is highly dependent on the length of spooled-out wire, shown in equation 4.2:

$$k_w = \frac{E \cdot A}{L} \tag{4.2}$$

where

E	-	Steel E-modulus
A	-	Wire cross-sectional area
L	-	Wire length

As internal damping in the steel wire is quite complex to model correctly, the value of the damping coefficient is set to be 10% of the value of the wire stiffness as a rule of thumb:

$$b_w = 0.1s \cdot k_w \tag{4.3}$$

An expression for total wire elongation is set up utilizing the angular position of the winch drum and vertical payload displacement, shown in equation 4.4, while wire elongation rate is presented in 4.5:

$$\delta = r_d \cdot \theta_m - z_{pl} + \delta_0 \tag{4.4}$$

$$\dot{\delta} = r_d \cdot \dot{\theta}_m - \dot{z}_{pl} \tag{4.5}$$

where

r_d	-	Winch drum radius
$\theta_m, \dot{\theta}_m$	-	Motor angular position and velocity
$z_{pl}, \dot{z}_{pl}$	-	Vertical payload position and velocity
δ_0	-	Static wire deflection

Static wire deflection is due to the forces acting on the suspended payload in equilibrium, as presented in equation 4.6. Figure 4.3 can be used for visual reference regarding equilibrium forces.

$$\delta_0 = \frac{\left(m_{pl} + \frac{m_w}{2}\right) \cdot g - F_{B,pl} - \frac{F_{B,w}}{2}}{k_w} \tag{4.6}$$

where

m_{pl}	-	Payload mass
g	-	Gravitational acceleration
$F_{B,pl}$	-	Payload buoyancy force
$F_{B,w}$	-	Wire buoyancy force

A free body diagram for the payload is presented in figure 4.3. For simplicity, the fraction of wire mass directly connected to the payload is included, adding to the payload mass. Figure 4.4 shows the kinetic diagram.

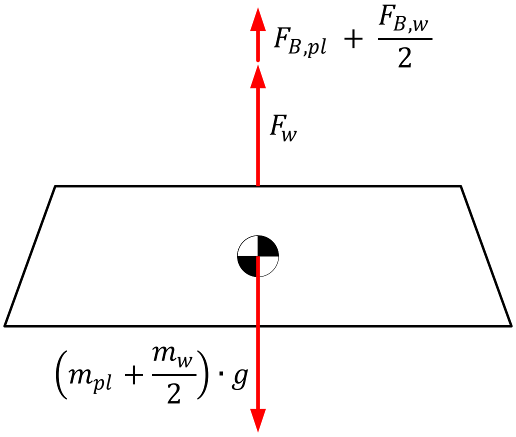

$F_{B,pl} + \frac{F_{B,w}}{2}$
F_w
$\left(m_{pl} + \frac{m_w}{2}\right) \cdot g$

Figure 4.3: Free body diagram of the payload

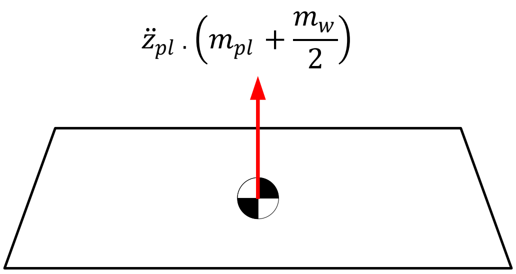


Figure 4.4: Kinetic diagram of the payload

The payload follows the equation of motion presented in 4.7, referred to the kinetic diagram, figure 4.4:

$$F_w + F_{B,pl} + \frac{F_{B,w}}{2} - \left(m_{pl} + \frac{m_w}{2}\right) \cdot g = \left(m_{pl} + \frac{m_w}{2}\right) \cdot \ddot{z}_{pl} \tag{4.7}$$

4.2 Winch Drum

This section covers the dynamic behaviour of the winch drum, in addition to effective drum inertia and radius calculations. The winch drum is directly driven by the hydraulic motor through a coupling, eliminating the use of a gearbox. Hence, no torque- or rotational speed levels are altered between the two components.

To derive the equation of motion for the drum, a free body diagram and a kinetic diagram is presented:

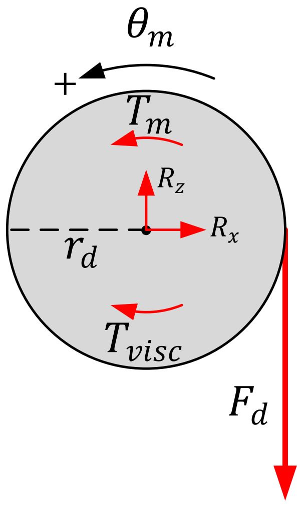


Figure 4.5: Free body diagram of the drum

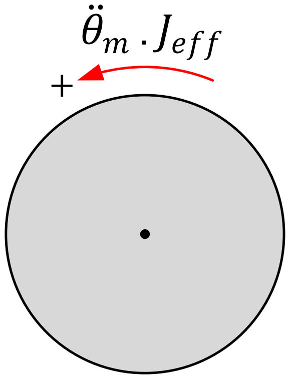


Figure 4.6: Kinetic diagram of the drum

The moment defined through the centre axis of the winch drum is

$$T_m - F_d \cdot r_d - T_{visc} = \ddot{\theta}_m \cdot J_{eff} \tag{4.8}$$

where

T_m	-	Torque acting on the drum from the motor
F_d	-	Total force acting on the drum
r_d	-	Drum radius
T_{visc}	-	Viscous friction moment
$\ddot{\theta}_m$	-	Drum angular acceleration
J_{eff}	-	Effective inertia on the motor

As the spring and damper is modelled in the middle of the wire, the total force acting on the drum must be the sum of the wire force derived in equation 4.1, the gravitational force acting of one half of the wire mass (as shown in figure 4.2) and its buoyancy:

$$F_d = F_w + \frac{m_w}{2} \cdot g - F_{B,\frac{w}{2}} \tag{4.9}$$

Some viscous friction moment T_{visc} is assumed to appear in the system and is included in equation 4.8. The friction moment is assumed to be proportional to the motor angular speed $\dot{\theta}_m$ by some constant b_b:

$$T_{visc} = b_b \cdot \dot{\theta}_m \tag{4.10}$$

4.2.1 Effective Radius

The considered winch drum has a capacity of $3660m$ of wire. Each layer spooled on increases the effective radius and thus the amount of wire per layer. A simplified sketch of the drum is shown in figure 4.7, where d_d is the diameter of the drum, d_w is the diameter of the wire and r_{eff} the effective radius:

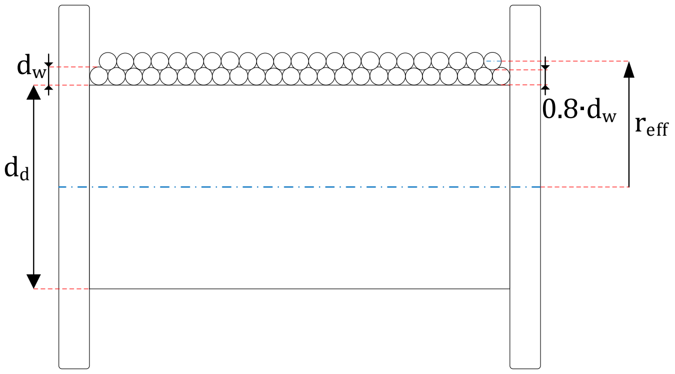


Figure 4.7: Drum parameters

As shown, each new layer after the first increases the radius by $0.8 \cdot d_w$, giving a general expression for the effective radius:

$$r_{eff} = \frac{d_d}{2} + d_w + (n_l - 1) \cdot 0.8 \cdot d_w - \frac{d_w}{2} \tag{4.11}$$

where r_{eff} is the effective radius, d_d the drum diameter, n_l the number of layers and d_w the wire diameter.

The effective radius is calculated by iteration in a MATLAB-script, as the radius changes for each layer of wire. The script first finds the maximum number of turns per layer n_t:

$$n_t = \left\lfloor \frac{w_d}{d_w} \right\rfloor \tag{4.12}$$

where w_d is the width of the drum and d_w the diameter of the wire, and n_t is rounded down to the nearest integer. The available number of turns changes with ±1 for each layer as the next wire layer is spooled in the grooves of the previous layer. The length of wire per turn is calculated:

$$L_{w,t} = 2\pi \cdot r_{eff} \tag{4.13}$$

The length of wire on the drum $L_{w,d}$ is then calculated. For each new layer, r_{eff} and $L_{w,t}$ is updated.

4.2.2 Effective Inertia

To find the effective inertia on the motor J_{eff} , a kinetic energy analysis is performed on the system. As the wire is modelled as a spring-damper and is not rigid, the payload is not included in the inertia analysis. However, since a considerable amount of wire is spooled out, half of the wire mass is used in the calculation of the drum inertia (as shown in figure 4.2):

$$\begin{aligned}
E_{kin,tot} = \frac{1}{2} J_{eff} \cdot \dot{\theta}_m^2 &= \frac{1}{2} J_m \cdot \dot{\theta}_m^2 + \frac{1}{2} J_d \cdot \omega_d^2 + \frac{1}{2} J_{w,d} \cdot \omega_d^2 + \frac{1}{2} \frac{m_w}{2} \omega_d^2 \cdot r_{eff}^2 \\
\frac{\cancel{1}}{\cancel{2}} J_{eff} \cdot \cancel{\dot{\theta}_m^2} &= \frac{\cancel{1}}{\cancel{2}} J_m \cdot \cancel{\dot{\theta}_m^2} + \frac{\cancel{1}}{\cancel{2}} J_d \cdot \cancel{\dot{\theta}_m^2} + \frac{\cancel{1}}{\cancel{2}} m_{w,d} \frac{d_d^2/4 + r_{eff}^2}{2} \cancel{\dot{\theta}_m^2} + \frac{\cancel{1}}{\cancel{2}} \frac{m_w}{2} \cancel{\dot{\theta}_m^2} \cdot r_{eff}^2 \\
J_{eff} &= J_m + J_d + m_{w,d} \frac{d_d^2/4 + r_{eff}^2}{2} + \frac{m_w}{2} r_{eff}^2 \qquad (4.14)
\end{aligned}$$

where:

$E_{kin,tot}$: Total kinetic energy
$\dot{\theta}_m$: Motor angular velocity
ω_d : Drum angular velocity
J_m : Motor rotor inertia
J_d : Drum moment of inertia
$J_{w,d}$: Moment of inertia of wire on drum
d_d : Drum diameter
r_{eff} : Drum radius including wire
$m_{w,d}$: Mass of spooled in wire
$m_{w,out}$: Mass of spooled out wire
m_{pl} : Payload mass
r_d : Drum radius

4.3 Load Analysis

This section presents an analysis of the load case in order to formulate demands used to set up requirements that the motor specifications must meet. The motor must be able to hold the design load and accelerate the effective inertia. This means that the highest available torque from the motor must meet the criterion

$$T_{m,max} > \ddot{\theta}_{m,dem} \cdot J_{eff} + T_{L,max} \qquad (4.15)$$

where $T_{m,max}$ is the maximum motor torque, $\ddot{\theta}_{m,dem}$ is the demanded acceleration on the motor shaft, J_{eff} the effective inertia and $T_{L,max}$ the maximum load torque, including the force on the drum F_d and friction moment T_{visc}.

4.3.1 Determined Parameters

In order to perform the analysis, some values not provided by neither the industrial winch supplier nor stated in the problem statement have been determined in order to have a fully defined model. These are presented in table 4.2:

Description	**Parameter**	**Dimension**
Payload volume	V_{pl}	$2.55m^3$
Maximum payload velocity	$\dot{z}_{pl,max}$	$1.5\frac{m}{s}$
Viscous friction coefficient	b_b	$1793Nm \cdot s$
Wave period	T_w	$9s$
Wave amplitude	A_w	$1m$

Table 4.2: Additional winch system parameters

Some parameters are dependent on further system specifications, such as the effective radius used for calculating the load torque, and the effective inertia that the motor must accelerate.

Effective Drum Radius

As the winch has a capacity of up to 3660 m of wire spooled onto the drum, the effective radius of the drum must be calculated. The payload will be simulated with 200 m of wire spooled out. The MATLAB script mentioned in chapter 4 ends up with 15 layers, and the wire is on its 29th turn (out of 37). The effective drum radius is then:

$$\begin{aligned} r_{eff} &= \frac{1632mm}{2} + 34mm + (15-1) \cdot 0.8 \cdot 34mm - \frac{34mm}{2} \\ r_{eff} &= 1213.8mm \end{aligned} \tag{4.16}$$

With all the wire spooled out, the effective radius is:

$$\begin{aligned} r_{eff} &= \frac{1632mm}{2} + 34mm + (1-1) \cdot 0.8 \cdot 34mm - \frac{34mm}{2} \\ r_{eff} &= 833mm \end{aligned} \tag{4.17}$$

Effective Inertia

To calculate the effective inertia that the motor needs to accelerate, equation 4.14 is used. The mass of the wire on the drum $m_{w,d}$ with the payload at $200m$ depth is

$$\begin{aligned} m_{w,d} &= \rho_w \cdot (L_{w,tot} - L_{w,out}) \\ m_{w,d} &= 5.66\frac{kg}{m} \cdot (3660m - 200m) = 19584kg \end{aligned} \tag{4.18}$$

where $L_{w,tot}$ is the total length of the wire and $L_{w,out}$ the length of the spooled out wire. The mass of the spooled out wire then:

$$m_{w,out} = 5.66\frac{kg}{m} \cdot 200m = 1132kg \tag{4.19}$$

The motor inertia J_m is assumed to be relatively low compared to the load, since there is no gearbox in between that reduces the inertia seen by the motor shaft. As a consequence, no calculation has been done as the only determined geometrical parameter for the motor is the displacement volume V_d. Instead, data from a traditional radial hydraulic motor has been used. The motor MRVE 8200 from Parker Calzoni with displacement $8226.4\frac{cm^3}{rev}$ has an inertia of $11376.6kgcm^2$ [12]. As the motor is expected to have a displacement around 70 000 cc, J_m is set to

$$J_m = 1.14kgm^2 \cdot \frac{70000\frac{cm^3}{rev}}{8226.4\frac{cm^3}{rev}} = 9.68kgm^2 \tag{4.20}$$

These values are put into equation 4.14 which yields:

$$\begin{aligned} J_{eff} &= 9.68kgm^2 + 1111kgm^2 + 19584kg\frac{\frac{(1.632m)^2}{4} + (1.2138m)^2}{2} + \frac{1132kg}{2}(1.2138m)^2 \\ J_{eff} &= 22901kgm^2 \end{aligned} \tag{4.21}$$

As expected, the motor inertia is much lower than the rest of the inertia in the system and no further estimation of J_m is deemed necessary.

4.3.2 Acceleration and Torque Demand

Two criteria dominate the acceleration demand: The payload must follow a varying velocity reference signal simulating an active heave compensation scenario, and it should ramp up the velocity to its maximum velocity ($\dot{z}_{pl} = 1.5\frac{m}{s}$) within a reasonable time t_r. As the wave compensation is the most critical criterion in regards to system accuracy and payload stability, it is chosen as the dimensioning factor for the system. Based on the data from table 4.2, the expressions for wave movement, velocity and acceleration are derived:

$$z_w = 1m \cdot \sin\left(\frac{2\pi}{9s} \cdot t\right) \tag{4.22}$$

$$\dot{z}_w = 1m \cdot \frac{2\pi}{9s} \cos\left(\frac{2\pi}{9s} \cdot t\right) \tag{4.23}$$

$$\ddot{z}_w = 1m \cdot -\left(\frac{2\pi}{9s}\right)^2 \sin\left(\frac{2\pi}{9s} \cdot t\right) \tag{4.24}$$

Where z_w is the vertical wave movement, $\dot{z}_w$ the wave velocity and $\ddot{z}_w$ the wave acceleration. The highest angular acceleration required by the motor to compensate for the wave motion occurs when the effective radius is smallest, i.e. when all wire is out:

$$\ddot{\theta}_{m,dem} = \frac{\left|1m \cdot -\left(\frac{2\pi}{9s}\right)^2\right|}{{}^{d_d}/_2 + {}^{d_w}/_2} = \frac{0.49\frac{m}{s^{-2}}}{0.833m} = 0.59s^{-2} \tag{4.25}$$

As the wire is flexible, the simplest way to find the highest load torque from the winch system at the demanded acceleration is to model it and simulate the acceleration of the drum with an ideal motor. The ideal motor replaces equation 4.8 (drum motion equation) in the model and gives an ideal acceleration profile up to the maximum payload velocity as shown in equation 4.26:

$$\ddot{\theta}_m = \begin{cases} \ddot{\theta}_{m,dem} & t < \frac{\frac{\dot{z}_{pl,max}}{r_{eff}}}{\ddot{\theta}_{m,dem}} \\ 0 & \text{else} \end{cases} \tag{4.26}$$

This acceleration profile is slightly tougher than the wave acceleration $\ddot{z}_w$, as $\ddot{z}_w$ is sinusoidal and not constant, thus yielding a slightly conservative torque demand. The acceleration is applied until the maximum payload velocity $\dot{z}_{pl,max}$. This value is higher than the maximum wave velocity $\dot{z}_w$, which also contributes to a slightly more conservative torque demand. The model is simulated with all the wire out, as this is the toughest condition (highest angular acceleration, highest spooled out wire mass, lowest wire stiffness). The script with the winch model with the ideal motor is shown in appendix A.1. The highest load torque recorded is:

$$T_{L,max} = 308\ 173 Nm \tag{4.27}$$

This yields a total torque demand for the motor:

$$\begin{aligned} T_{dem} &= T_{L,max} + \ddot{\theta}_{m,dem} \cdot J_{eff} \\ T_{dem} &= 308\ 173 Nm + 0.59 s^{-2} \cdot 8308 kg \cdot m^2 \\ T_{dem} &= 313\ 034 Nm \end{aligned} \tag{4.28}$$

Note that the effective inertia used here is computed for the scenario with all wire spooled out, and not the one computed in section 4.3.1. The demanded torque can be used to formulate required specifications for the motor cylinders and valves.

5 | Modelling and Simulation

This chapter presents calculation of necessary system requirements, in addition to explaining the modelling of main system components. Modelling of the system is based on the respective theory sections, with the digital valves and valve dynamics presented in more detail. The following subsystems are modelled and implemented in MATLAB simulations:

- The digital motor fundamentals, presented in chapter 2
- The control system, based on chapter 3
- The mechanical winch system, presented in chapter 4

The MATLAB simulations uses a fixed time-step forward Euler method, with the fixed step-size of $0.005ms$. Before the model of the complete system can be simulated, necessary requirements for the components musts be calculated. Since the main focus of this project is secondary control of the digital motor, a detailed model of the ring-line hydraulic power units (HPUs) are considered outside of the project scope. A detailed model of the hydraulic system is not a focus in this report. Therefore, the lines connected to the digital motor is modelled as constant pressure sources. The high-pressure line is modelled as a constant pressure source of $p_H = 350bar$, and the low-pressure line as a constant pressure source of $p_L = 5bar$. No energy storage system is considered, such as an accumulator setup, despite the possibility to store high-pressure fluid generated from pumping operations.

5.1 Required Motor Displacement

Based on the load analysis presented in section 4.3, the required motor displacement can be determined. For a hydraulic motor, the highest output torque can be expressed as in equation 5.1:

$$T_{m,max} = \frac{D_m \cdot \Delta p_{motor}}{2\pi} \tag{5.1}$$

Where $T_{m,max}$ is the maximum motor torque, D_m the total motor displacement volume and Δp_{motor} the pressure drop across the motor. Total motor displacement volume is a product of number of cylinders, n_c, and the displaceable cylinder volume, V_d:

$$D_m = n_c \cdot V_d \tag{5.2}$$

Since choosing the optimal combination of these two design parameters is outside the scope of this project, number of cylinders is set to 42. Theoretically, a 42-cylinder motor can be assembled by e.g. stacking six seven-cylinder motors in series, with each motor shifted by an angle that gives the same $\Delta\theta$ between all cylinders. This leaves the cylinder displacement V_d

as the only factor that determines the output torque. Thus, the required cylinder displacement volume can be determined by rearranging equation 5.1 and setting $T_{m,max} = T_{dem}$, obtained from equation 4.28.

$$\begin{aligned} V_d &= \left\lceil \frac{2\pi \cdot T_{dem}}{n_c \cdot \Delta p_{motor}} \right\rceil \\ V_d &= \left\lceil \frac{2\pi \cdot 313\ 034 Nm}{42 \cdot \sqrt{345 \cdot 10^5 Pa}} \right\rceil \\ V_d &= 1400cc \end{aligned} \tag{5.3}$$

The motor parameters used in the simulation model are collected in table 5.1:

Description	Parameter	Value	Unit
Number of cylinders	n_c	42	$[-]$
Number of valves	n_v	84	$[-]$
Cylinder displacement volume	V_d	1400	cc

Table 5.1: Motor parameters

5.2 Valve Requirements

This section presents how the two valve parameters, the flow-pressure coefficient k_f and the valve switching time t_s, are determined. The procedure for calculating these requirements is described in section 2.3.1.

The valves should be fast enough and able to let through a required flow at a given pressure difference. Also, to select an efficient valve for the winch application, different system parameters must be evaluated. In this case, the DDM is operated using partial stroke displacement, which means that the valves are switched at less favourable conditions compared to full stroke displacement.

Referring to figure 2.8, the most challenging case of valve switching is at 25% displacement. This corresponds to the bottom left graph. An efficiency of $\approx 97\%$ is obtainable at $\bar{t}_s = 1\%$, which is feasible due to low rotational speeds of the considered winch system. At $\bar{t}_s = 1\%$, the corresponding reading of the pressure difference ratio is approximately 0.35%, needed for calculating the valve flow-pressure coefficient, k_f.

Considering the highest angular velocity scenario, with $\dot{z}_{pl} = 1.5\frac{m}{s}$ and $r_{eff} = 0.833m$, maximum angular velocity can be determined by:

$$\dot{\theta}_{m,max} = \frac{1.5\frac{m}{s}}{0.833m} = 1.8\frac{rad}{s} \tag{5.4}$$

Using the obtained normalized switching time $\bar{t}_s = 1\%$, the required switching time $t_{s,dem}$ is calculated based on equation 2.12:

$$\begin{aligned} t_{s,dem} &= 1\% \cdot \frac{2\pi}{\dot{\theta}_{m,max}} \\ t_{s,dem} &= 0.01 \cdot \frac{2\pi}{1.8\frac{rad}{s}} \\ t_{s,dem} &= 35ms \end{aligned} \tag{5.5}$$

As expected, a low speed DDM does not require very fast valves. Instead, the flow-pressure coefficient k_f is the dominating requirement for the valve. Maximum allowed pressure drop over the valve, $\Delta p_{valve,max}$, and maximum required flow, Q_{max}, must be computed to obtain the mentioned valve efficiency of $\approx 97\%$:

$$\frac{\Delta p_{valve,max}}{\Delta p_{machine}} = 0.35\% \tag{5.6}$$

$$\begin{aligned} &\Downarrow \\ \Delta p_{valve,max} &= 0.0035 \cdot (350bar - 5bar) \\ &= 1.21bar \end{aligned}$$

Also, maximum flow through the valve must be computed. Equation 2.14 and 2.16 are used to express the flow requirement per cylinder for the highest angular velocity scenario:

$$\begin{aligned} Q_{avg} &= \dot{\theta}_{m,max} \cdot V_d \\ Q_{avg} &= 1.8\frac{rad}{s} \cdot \frac{60\frac{s}{min}}{2\pi\frac{rad}{rev}} \cdot 1.4\frac{l}{rev} = 24.07\frac{l}{min} \end{aligned} \tag{5.7}$$

Maximum flow in the cylinder is higher, as described in section 2.3:

$$\begin{aligned} Q_{max} &= \pi \cdot Q_{avg} \\ Q_{max} &= \pi \cdot 24.07\frac{l}{min} = 76\frac{l}{min} \end{aligned} \tag{5.8}$$

The orifice equation (equation 2.9) is then used to find a flow-pressure coefficient k_f that allows a flow of Q_{max} at the maximum allowable pressure drop, $\Delta p_{valve,max}$. The expression is evaluated with the valve fully open, $\bar{x} = 1$:

$$\begin{aligned} Q &= \frac{1}{k_f} \cdot \sqrt{\Delta p_{valve}} \\ &\Downarrow \\ k_f &= \frac{\sqrt{\Delta p_{valve,max}}}{Q_{max}} \\ k_f &= \frac{\sqrt{1.21 \cdot 10^5 Pa}}{\frac{76}{60 \cdot 10^3}\frac{m^3}{s}} = 275\ 961\frac{s \cdot \sqrt{Pa}}{m^3} \end{aligned} \tag{5.9}$$

Finally, an evaluation of the available valves is performed, to analyse feasibility of the established requirements. Paper [7, p. 21] presents an overview over relevant valves, where some of the considered valves have a pilot stage. The pilot stage is needed in order to open the valve against a high pressure difference, Δp_{valve}, due to the selected displacement strategy in section 3.1.2. The MOOG N-DSHR-E40 pilot stage seat valve presented in the paper features a switching time of $t_s = 12ms$ and flow capacity of $1450\frac{l}{min}@5bar$. This is the smallest MOOG valve of this type, which is excessively large for the required specifications in this project. It is therefore feasible to assume a scaled-down version of this valve to feature the parameters collected in table 5.2, which are further used for simulation purposes. In addition, an arbitrary charge time t_{charge} is defined to account for any valve actuator delay.

Description	**Parameter**	**Value**	**Unit**
Switching time	t_s	8	$[ms]$
Actuator charge time	t_{charge}	30	$[ms]$
Flow-pressure coefficient	k_f	275 961	$[\frac{s\cdot\sqrt{Pa}}{m^3}]$

Table 5.2: Summarised Valve Parameters

5.3 Variable Bulk Modulus

As presented in section 2.4, the bulk modulus is a variable parameter. In order to improve the quality and accuracy of the simulation model, a variable bulk modulus is implemented to ensure correct pressure calculations. The process is assumed to be adiabatic, meaning no heat exchange with the surroundings. The oil temperature is considered constant during operation. The gas percentage in the oil is arbitrary set to 0.2 %, and the bulk modulus of the hydraulic fluid is set to 16 $000bar$, which is a typical value for an industrial hydraulic fluid. This gives the following parameters used in the simulation for the bulk modulus:

Description	**Parameter**	**Value**	**Unit**
Liquid bulk modulus	β_L	16000	$[bar]$
Percentage of gas in liquid	ϵ_g	0.002	$[-]$
Adiabatic coefficient	κ	1.4	$[-]$

Table 5.3: Variable bulk modulus

5.4 Valve Dynamics

This section presents how the digital valves are modelled and implemented in the simulations. Dynamic behaviour and valve requirements are set based on the presented theory in section2.3. Due to the selected displacement strategy, PSS, the valves are actively opened and closed in the simulations. In detailed valve studies, a point of interest could be to improve valve durability and reliability, for example by minimizing the impact force when the valve closes. Here, valve acceleration and velocity are important properties. However, a detailed valve study is outside the scope for this thesis. The only necessary movement related parameter is the valve plunger position, which enables calculation of flow through the valves, as shown in orifice equation 5.10. The sign function is included to determine the direction of flow.

$$Q = \frac{\bar{x}}{k_f} \cdot \sqrt{|\Delta p_{valve}|} \cdot sign(\Delta p_{valve}) \tag{5.10}$$

where

Q	-	Flow through the respective valve
$\bar{x}$	-	Normalized valve position
Δp_{valve}	-	Pressure drop across the valve
k_f	-	Flow pressure coefficient

Therefore, it is desired to reduce computational demand for calculating valve positions, by simplifying the valve dynamics described in section 2.3. The presented method uses a specific acceleration profile that must be time integrated to produce valve position data, whenever a valve is actuated. This method is simplified by replacing the continuous integration with a simple polynomial, which closely approximates the valve plunger position profile. Equation 5.11 presents the form of which the polynomial is constructed:

$$\bar{x}(t) = p_0 \cdot t^n + p_1 \cdot t^{n-1} + p_2 \cdot t^{n-2} + ... + p_{n-1} \cdot t + p_n \tag{5.11}$$

By simulating the valve *once* in a separate simulation, and performing time integration from acceleration to position, produced valve position data can be stored. Then, the stored data can be used to form the polynomial. MATLAB *polyfit* polynomial curve fitting was used to return the best fit, resulting the 10th degree polynomial vector found in equation 5.12.

$$\begin{aligned}\vec{p} = [&-1.0292e19 \quad 2.2409e21 \quad -8.0663e19 \quad 1.1955e18 \quad -9.3825e15 \quad 4.1885e13 \\ &-1.0740e11 \quad 1.5274e8 \quad -78551.6733 \quad 32.4710 \quad -0.0023]\end{aligned} \tag{5.12}$$

Figure 5.1 shows the constructed 10^{th} degree polynomial that describes normalized valve plunger position with respect to time. Also, the original, time integrated valve plunger position profile is shown for comparison. The polynomial approximates the time integrated plunger position well, with maximum residual recorded to $0.0028 \cdot 10^{-3}$.

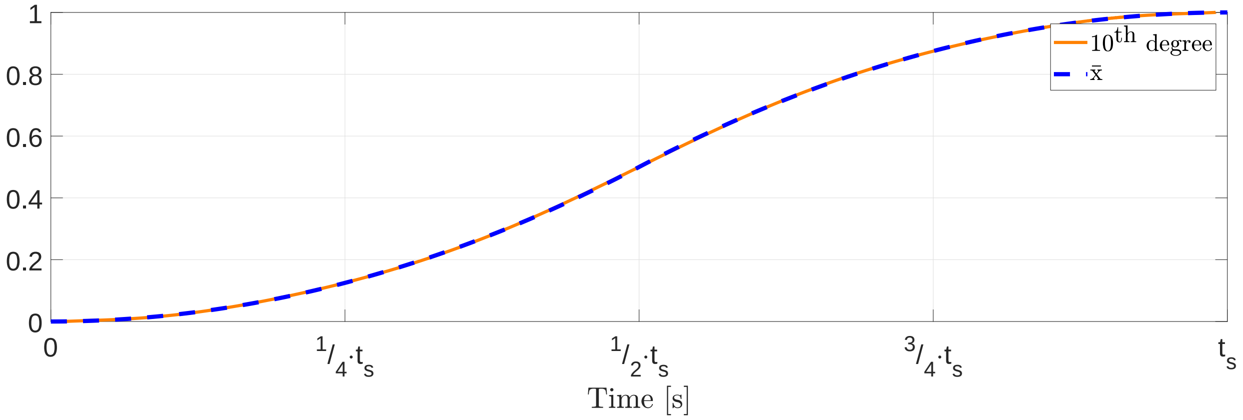


Figure 5.1: Original and approximated valve position profile

By using this method, all valve positions can be obtained by evaluating the same polynomial with respect to time, rather than performing time integrations on $n_v = 84$ acceleration profiles. The simulation model calculates the valve positions in the following way:

- All valves are assigned a local time variable
- All valve plunger positions are obtained by evaluating the same polynomial, using the respective valve time variable
- Time span ranging from $t = 0$ (fully closed) to $t = t_s$ (fully open)
- When a valve receives the opening control-signal, time starts to increment from $t = 0$ towards t_s, gradually opening the valve
- When a valve recieves the closing-signal, time starts to decrement from $t = t_s$ towards $t = 0$, gradually closing the valve
- The valve can not change its direction during an opening or closing sequence
- A time delay t_{charge} is added between each actuation sequence, representing necessary charge time for the valve actuator

Figure 5.2 presents a generic opening and closing sequence for a HPV. As mentioned above, the sequence is initiated by applying a control signal. Here, this is represented by the boolean variable $HPVb$, where a value of 1 is the opening control signal and 0 is the closing signal. As can be seen, once the opening signal is given, the valve opening sequence is initiated. In this particular case, the signal given by the control system is reset to zero almost immediately, meaning that the valve should be closed again. However, the valve should not be able to respond to this command before the mentioned charge time ,t_{charge}, has elapsed. Therefore, an additional boolean variable $HPV_{opening}$ is used to ensure that both the actuation sequence and charge time is completed before a new sequence can take place.

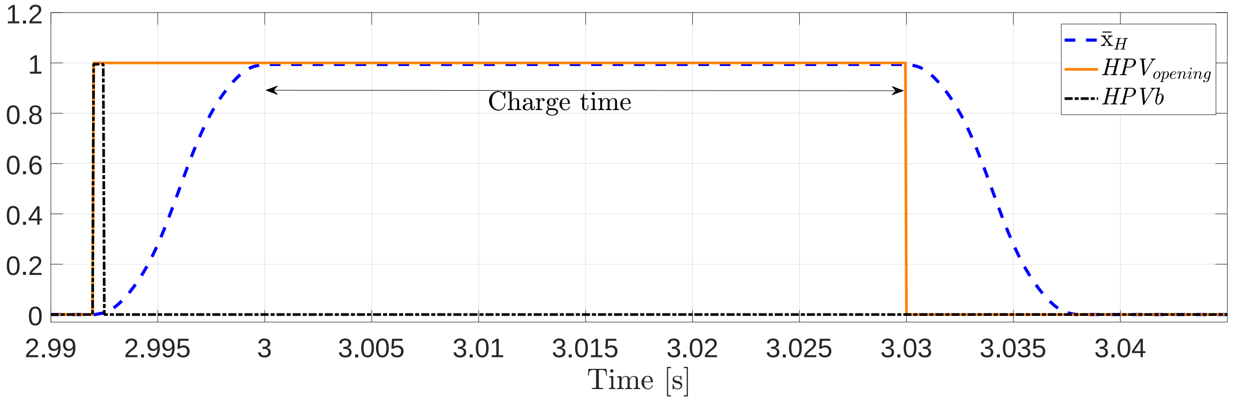


Figure 5.2: Generic HPV opening and closing sequence

Now that the valve dynamics and modelling philosophy is explained, it should be mentioned that the same procedure is used for the low-pressure valves (LPVs).

5.5 Feed Forward Signal for Velocity Controller

As the DDM is considered for heavy duty winch application, it is reasonable to assume that the mass of the attached payload is a known parameter. A feed forward element in the controller is therefore implemented that helps counter the static load. The motor is simulated with α incrementing by 0.01 rad every revolutionary period, and the torque output from the motor is recorded, shown in figure 5.3. In a steady state situation, the torque from the motor should be equal to the load torque. Hence, the data acquired can be used to express a given α to counter the load torque. Figure 5.4 plots the same data, but with α as a function of T_L where $T_L = T_m$.

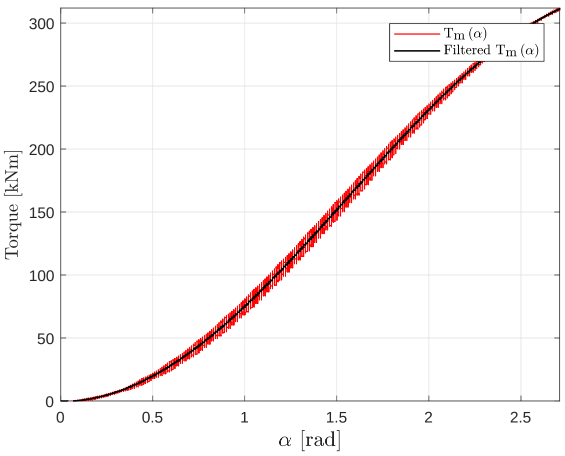


Figure 5.3: Relationship between motor torque and displacement parameter α

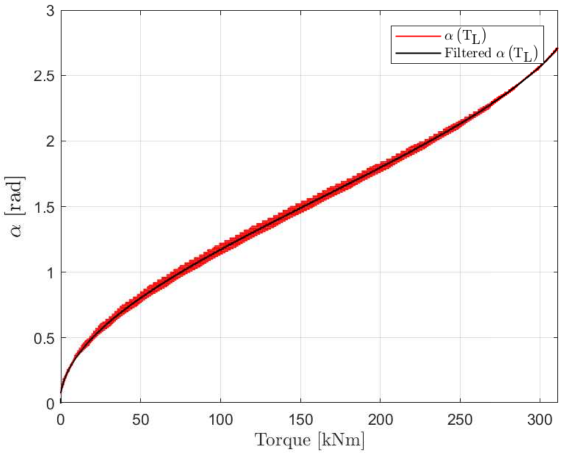


Figure 5.4: Displacement parameter α as a function of load torque T_L

The filtered output (black line) from figure 5.4 can be used as a reasonable approximation value for a feedforward parameter α_{FF}. The filtered data is approximated by a MATLAB tool with a polynomial expression in order to create a function that takes in a load torque and suggests a feedforward parameter α_{FF}. The expression takes the following form:

$$\begin{aligned}
\vec{p} = [&-3.9755e{-52} \quad 6.5394e{-46} \quad -4.6192e{-40} \quad 1.8320e{-34} \quad -4.4776e{-29} \\
&6.9703e{-24} \quad -6.9086e{-19} \quad 4.2534e{-14} \quad -1.5666e{-09} \quad 4.0771e{-05} \quad 0.0727] \\
\alpha(T_L) &= p_0 \cdot T_L^1 0 + p_1 \cdot T_L^9 + p_2 \cdot T_L^8 + ... + p_9 \cdot T_L + p_1 0
\end{aligned}$$

6 | Validation of Control System

This chapter presents the validation of the control system suggested in section 3.3. The presented results are focused on cases where the control system changes operation mode, as this is an area where challenges are expected. Tests are performed to check the control system's ability to switch between the different modes of the DDM, defined in section 3.3.1. The tests are performed with a simplified load that can easily be manipulated in combination with the reference velocity in order to force the control system to change operation mode.

The developed control system chooses the operating mode based on the flow chart (figure 3.23) shown in section 3.3. As the DDM is considered for a winch application, focus has been aimed at ensuring smooth transition between the first and the third quadrant. This is because a winch is expected to have a gravitational load, restricting the operation of the motor to mostly these two quadrants. In addition, some transition through the second quadrant is expected for the lowering of a light load. Handling of a light load is mentioned in the problem statement and must therefore be taken into consideration to some extent.

The tests have been conducted with maximum velocities of $10rpm$, corresponding to approximately $1.05\frac{rad}{s}$. A simplified load with a maximum value of $T_L = 0.8 \cdot T_{max}$ has been used, which yields approximately $258\ 290Nm$. Controller parameters used for the tests are collected in table 6.1

Parameter	**Value**
K_p	1.34
T_i	0.86
T_d	0.015

Table 6.1: Hoisting and lowering PID Parameters

6.1 First to Third Quadrant

This section presents the results from the mode switch validation from the first to the third quadrant. As depicted in figure 3.13, the load torque is negative (works in CW-direction) for both quadrants 1 and 3. The velocity is positive (CCW-direction) for the 1st quadrant, and negative (CW-direction) for the 3rd quadrant. Thus, to force a mode change from 1st to 3rd quadrant, the load torque is kept constant and the direction of the velocity reference is changed from a positive value to a negative value.

Figure 6.1 shows a plot of the velocity reference and the velocity of the motor shaft. The light blue line in the middle of the plot indicates the change in operation mode from 1st quadrant to 3rd quadrant. An irregularity in the motor shaft velocity plot is observed at $t \approx 13s$.

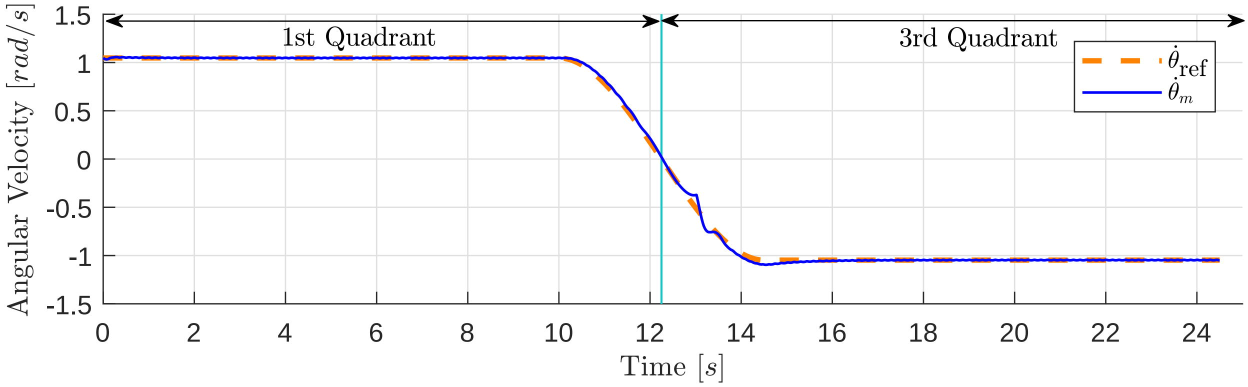


Figure 6.1: Shaft velocity and reference

Figure 6.2 shows how the load torque is held constant (orange line), and the blue line shows the motor torque.

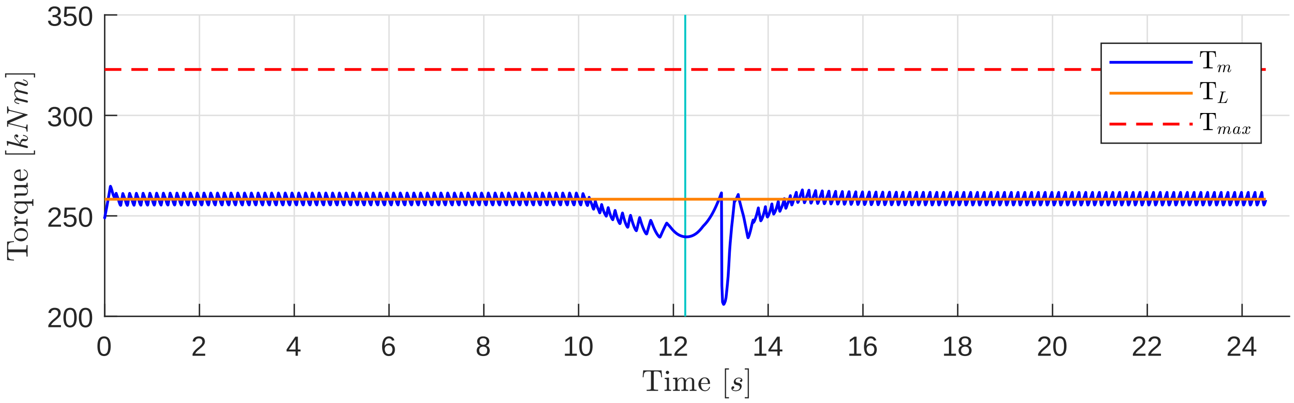


Figure 6.2: Motor torque and load torque

Figure 6.3 shows the control parameter α in radians.

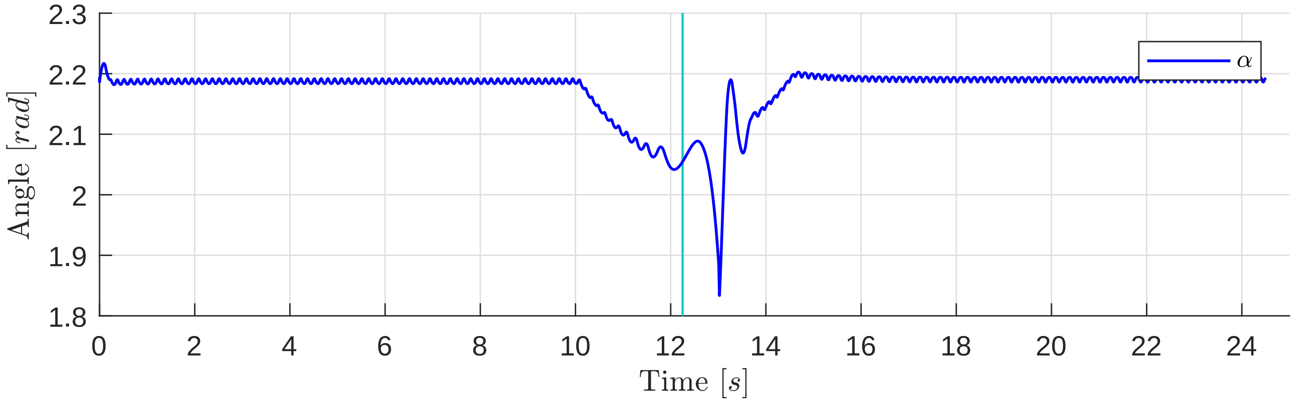


Figure 6.3: Control parameter α

Figure 6.4 shows a typical cylinder behaviour (cylinder 2) through the simulation. The green line shows the cylinder angle θ_c, the black line shows the chamber pressure p_c, the dashed blue line the LPV plunger position x_L and the dashed red line the HPV plunger position x_H. All values are normalized, i.e. a value of 1 simply means that the plotted variable is equal to its reference value. For the valves plungers, this means a fully open orifice. The pressure reference is at $350bar$, which is the same level as the high pressure line. The reference value for the cylinder angle θ_c is 2π, as it runs from 0 to 2π. This means that both 0 and 2π represent TDC. When plotting the cylinder angle, it is easier to detect a change in velocity direction than for the piston movement. The dashed grey horizontal line at 0.5 is added to help the reader follow when the angle reaches π, i.e. the cylinder BDC.

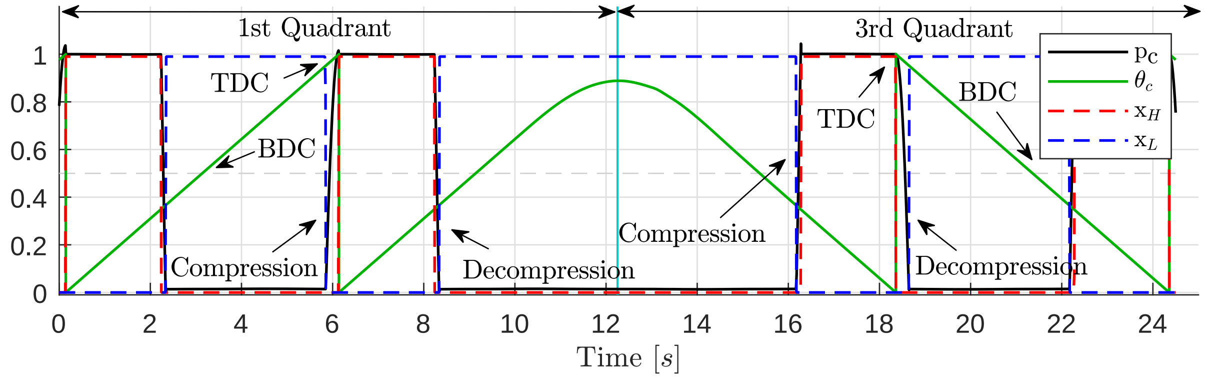


Figure 6.4: Behaviour of cylinder 2

A typical first quadrant cycle is observed from start to $t \approx 6s$. A pressurised cylinder performs work on the motor shaft until the cylinder angle reaches the parameter α decided by the control system, where the cylinder is decompressed. The cylinder then displaces low pressure fluid on its way back to TDC, where the pressurisation sequence is initiated. This particular cylinder in its passive period when the mode is switched. After the mode switch (right after $12s$), the cylinder crosses BDC and starts the compression sequence at an angle decided by the control parameter α. The load then does work on the cylinder, until it reaches TDC and the decompression sequence is initiated.

Figure 6.5 shows the flows through the HPV and LPV for the same cylinder. The small peaks are due to the simplifications introduced in the valve timing calculations discussed in section 2.6. The valves are thus opened with pressure differences slightly higher than calculated, as can also be seen by observing the pressure plot in figure 6.4, at for example $t \approx 16s$. The direction of the flows through the valves are reversed when the motor shaft changes direction, and are therefore negative after the mode switch.

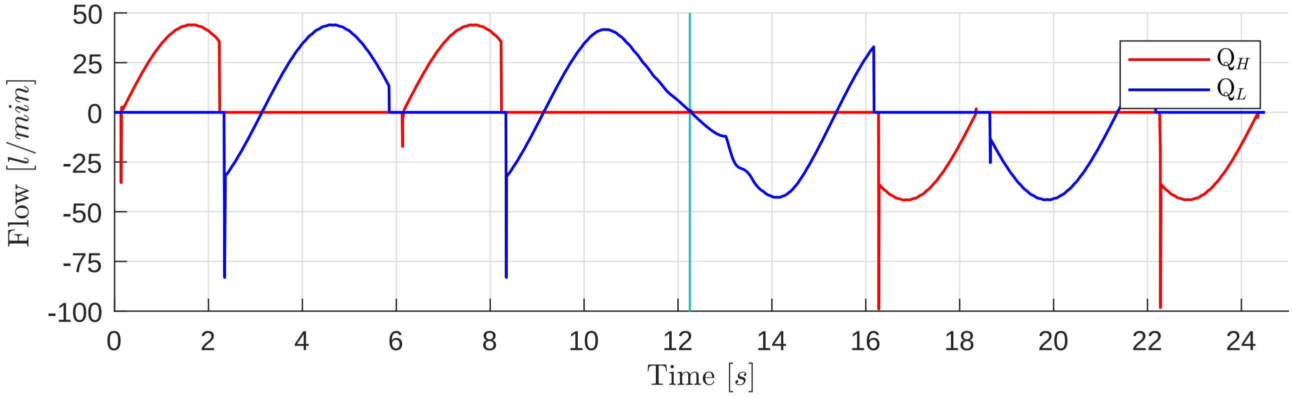


Figure 6.5: Flows through HPV and LPV of cylinder 2

To check if all cylinders are operating as desired, the total flows into the motor and out of the motor are plotted in figure 6.6. High peaks in these plots might indicate a problem during the mode switching.

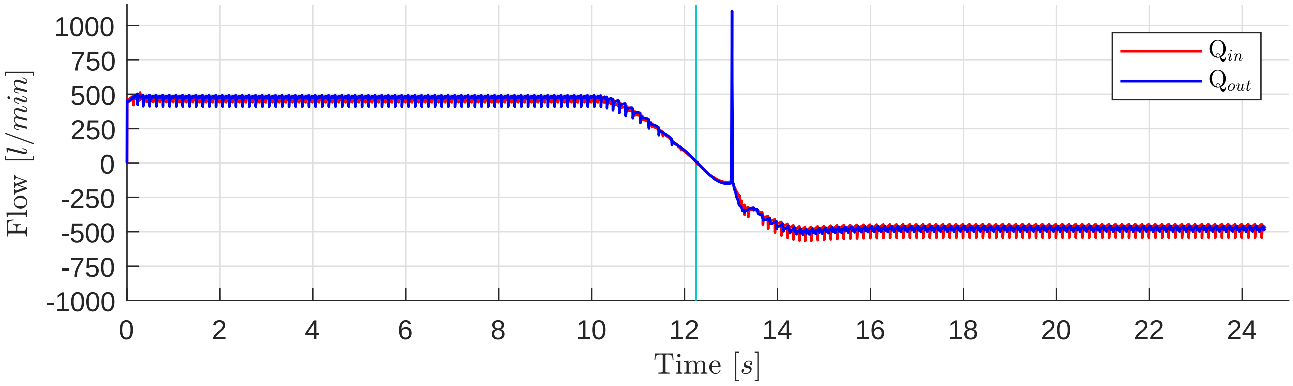


Figure 6.6: Total flow in and out of the motor

The peak at $t \approx 13s$ must be further addressed, as it might be a consequence of the mode switch. The cylinder is identified as cylinder 25, shown in figure 6.7.

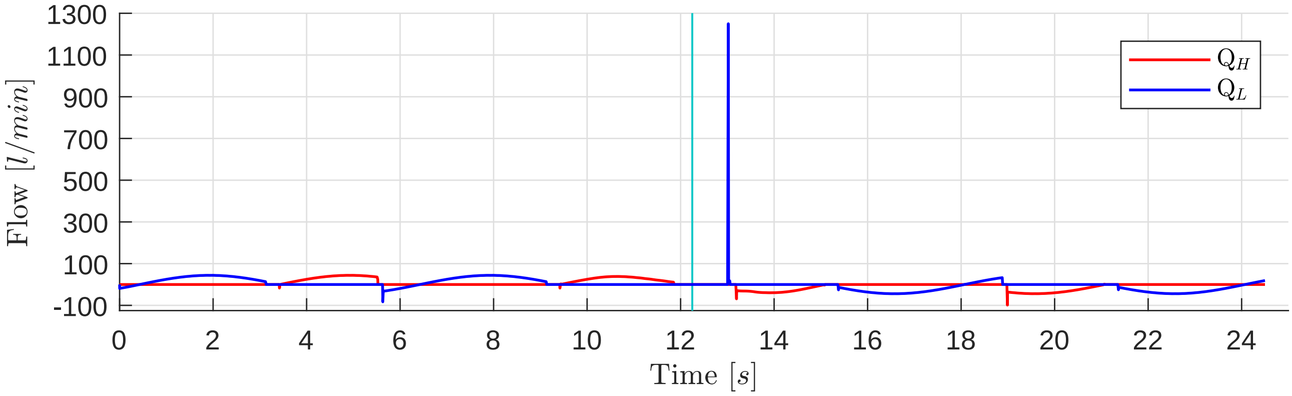


Figure 6.7: Flows through HPV and LPV of cylinder 25

The cause of the flow peak can be determined by examining the behaviour of the cylinder. Figure 6.8 shows that the cylinder was in the middle of a decompression sequence when the motor shaft velocity changed direction. As a consequence, the cylinder chamber is instead pressurised as the shaft now rotates in the negative direction and rises to levels of more than two times that of the high-pressure line.

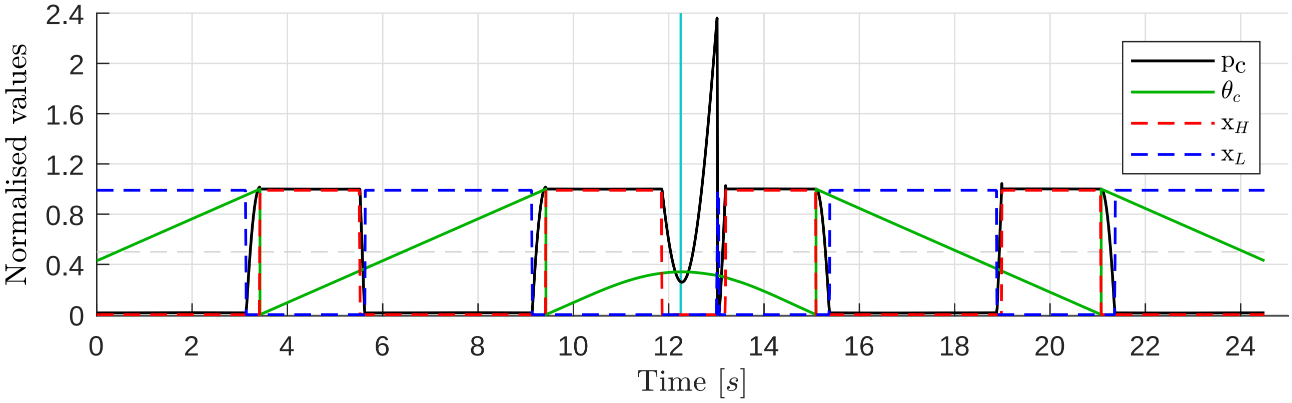


Figure 6.8: Behaviour of cylinder 25

This scenario clearly represents an operational hazard as the motor cylinders are most likely not dimensioned for these pressure levels. A special condition is therefore added to the control system to counter this. As the piston travels back up again, the HPV is allowed to open at the same angle as it closed in the first quadrant. This ensures that no uncontrolled pressure build-ups occur. Figure 6.9 plots the same cylinder (cylinder 25), now with the new statement added to the control system:

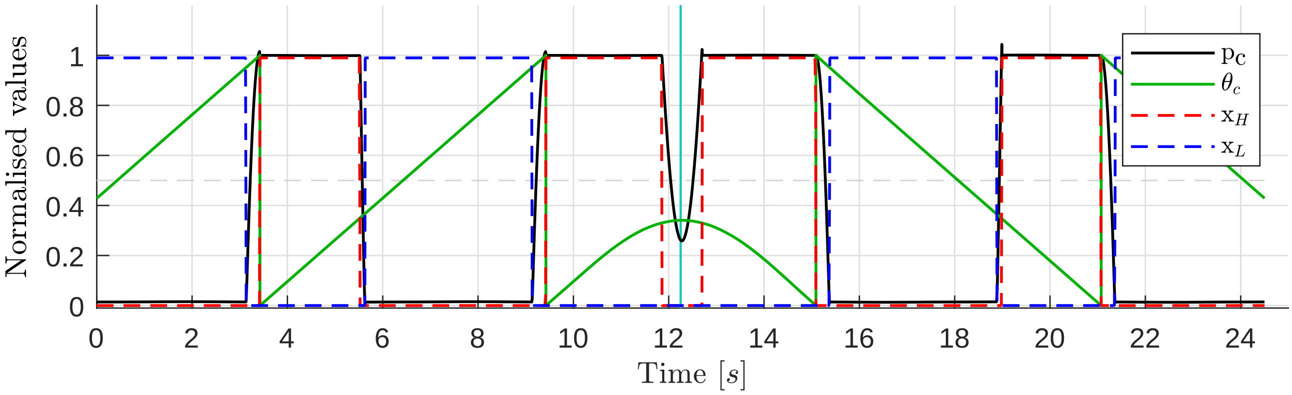


Figure 6.9: Cylinder 25, with improved mode switching

The pressure level is now controlled. Thus, the flows in the cylinder behave normally, as depicted in figure 6.10.

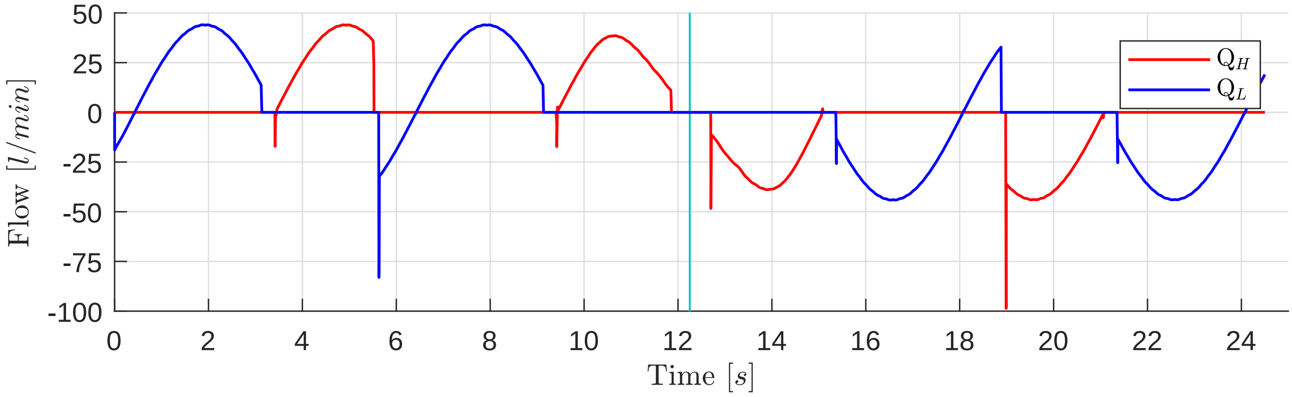


Figure 6.10: Flows in cylinder 25, with improved mode switching

For reference, cylinder 41 is plotted in figures 6.11 and 6.12 to show how a cylinder behaves when the mode switch is done during compression of a cylinder chamber. As the direction of the motor shaft is reversed, the fluid decompresses. The LPV then opens over a low pressure difference. This is due to the similar valve timing strategy of the 1st and the 3rd quadrant around TDC. When studying the valve diagrams of the two quadrants in section 3.3.1 (figures 3.16 and 3.20), it can be seen that the LPV closing angle for the compression in the 1st quadrant is close to the LPV opening angle for the decompression in the 3rd quadrant. Thus, even though the valve timing scheme has been changed, the LPV opens over a low pressure difference.

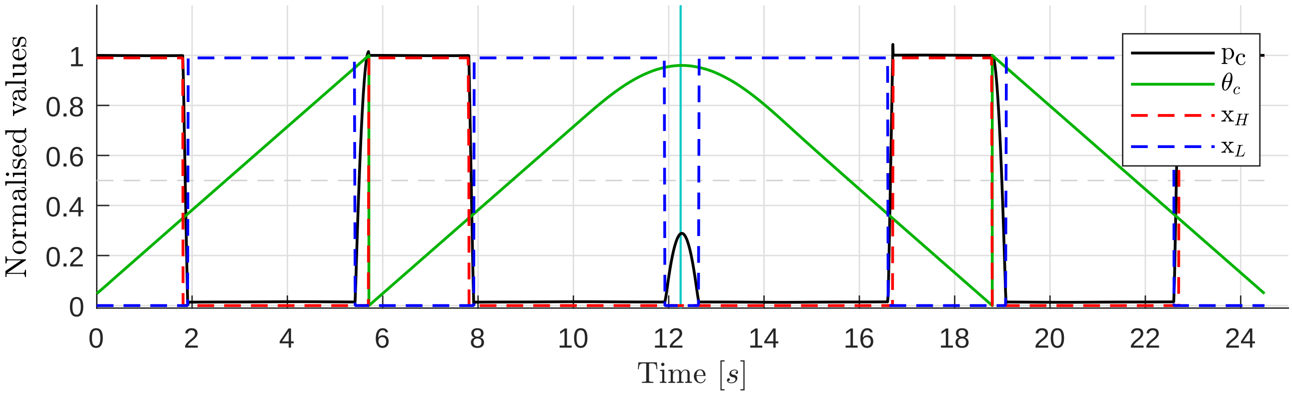


Figure 6.11: Behaviour of cylinder 41

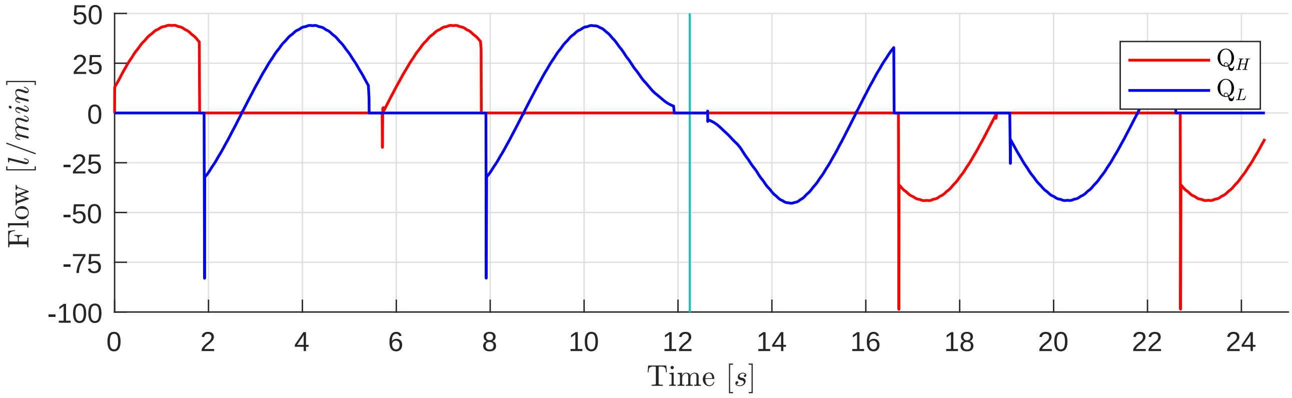


Figure 6.12: Flows over HPV and LPV for cylinder 41

Now that no undesired pressure build-up occurs in the cylinders when the mode is switched, the motor velocity has a better reference tracking compared to figure 6.1:

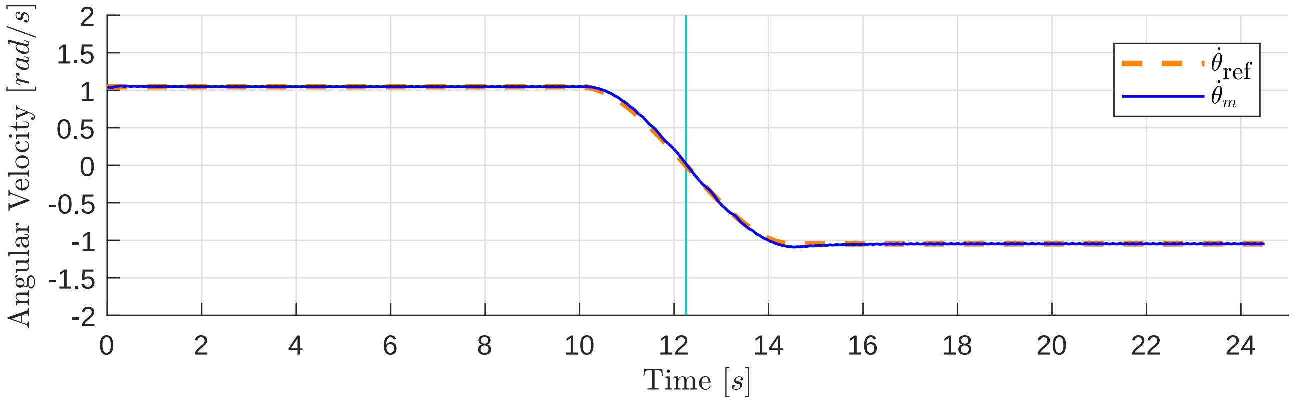


Figure 6.13: Shaft velocity and reference with improved mode switching

The plot of the displacement parameter α (figure 6.15) and the resulting torque (figure 6.14) are smoother as a consequence.

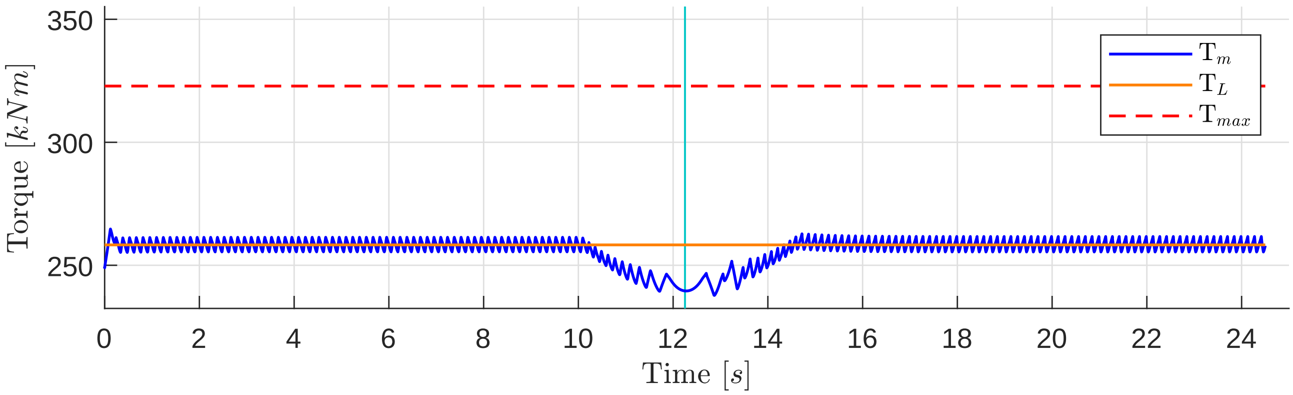


Figure 6.14: Motor torque, with improved mode switching

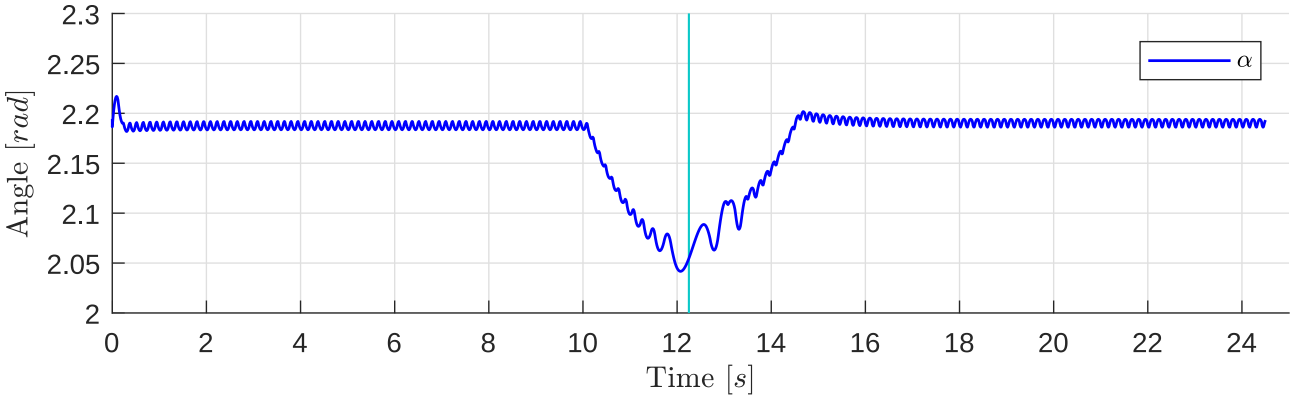


Figure 6.15: Control parameter α, with improved mode switching

The total flows in and out of the motor are now plotted again to ensure that the motor is operating as expected:

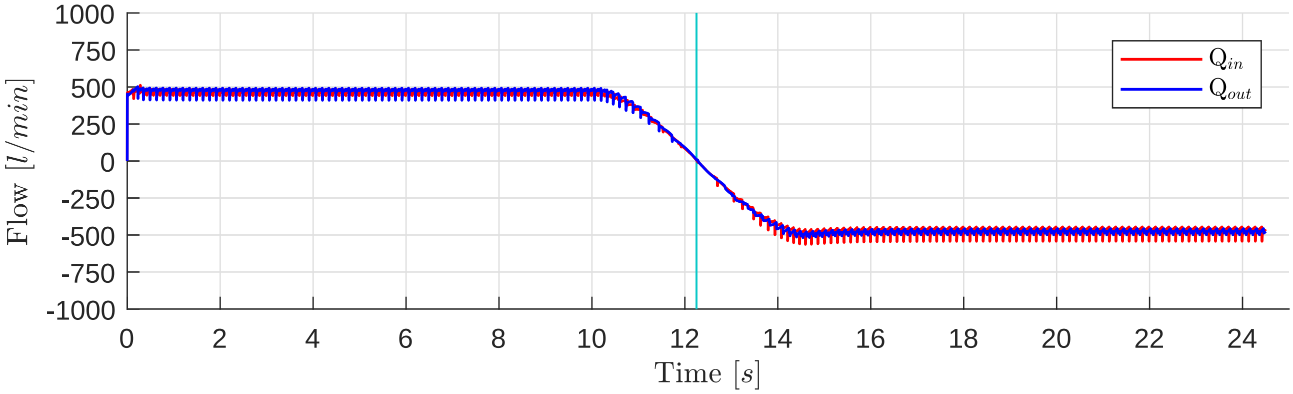


Figure 6.16: Total flow in and out of the motor

No peaks are observed, hence no further actions are taken.

6.2 Third to First Quadrant

This section presents the results from the mode switch validation from the third to the first quadrant. To force a mode change from the first to the third quadrant, the torque is kept constant and the direction of the velocity reference is changed from a negative value to a positive one. Figure 6.17 shows a plot of the velocity reference and the velocity of the motor shaft. The light blue line in the middle of the plot indicates the change in operation mode from the third to the first quadrant.

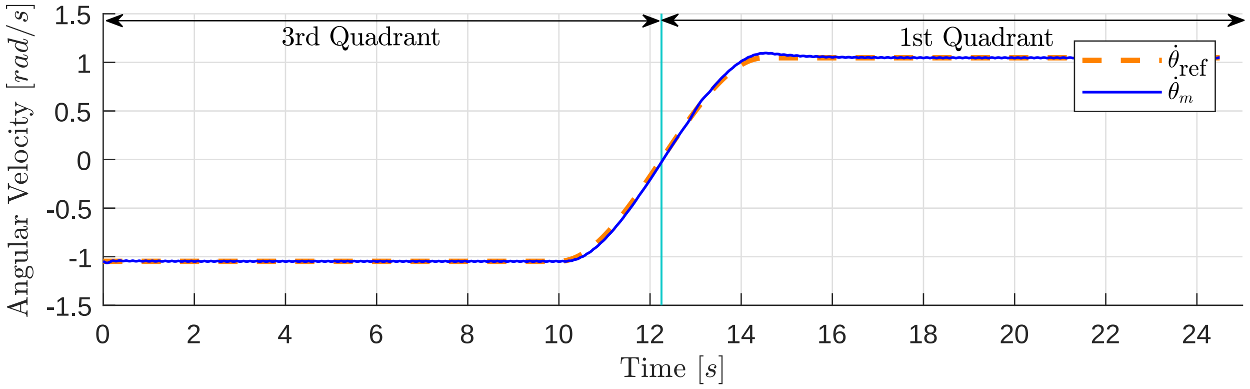


Figure 6.17: Shaft velocity and reference

Figure 6.18 plots the load torque and the motor torque. The load torque is the same for both the 1st and the 3rd quadrant and is therefore kept constant.

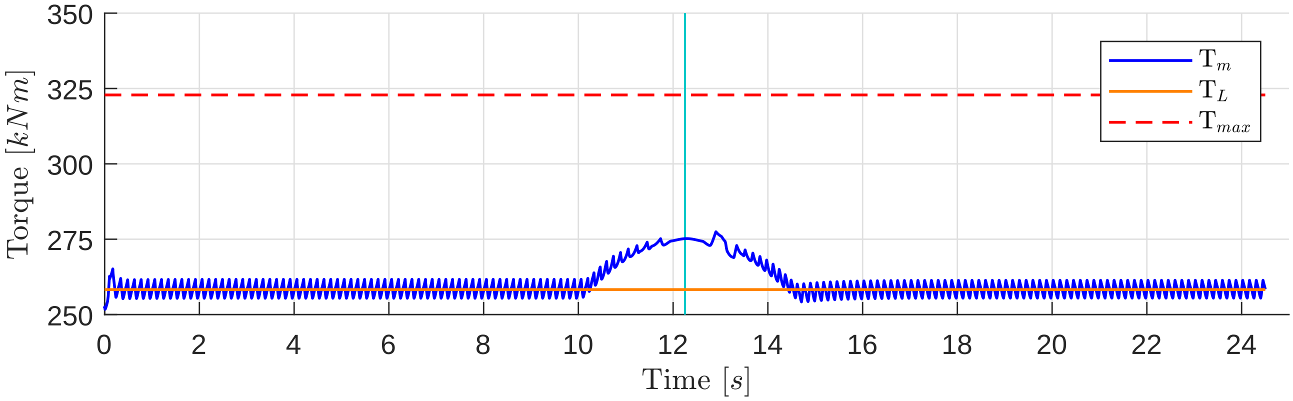


Figure 6.18: Motor torque and load torque

Figure 6.3 shows the control parameter α in radians.

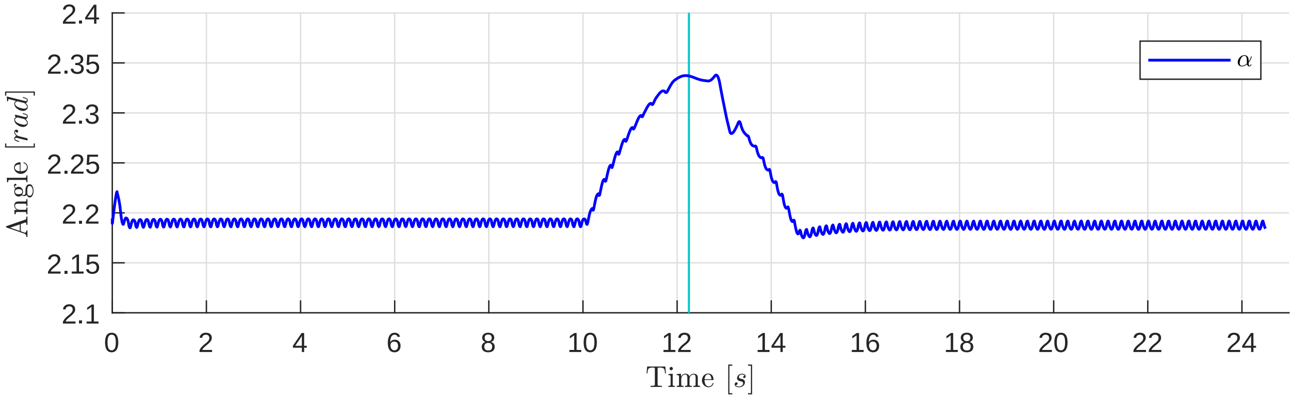


Figure 6.19: Control parameter α

Figure 6.20 shows a random cylinder (cylinder 22) during the simulation with common behaviour. The mode switch occurs right after $t \approx s$, during which the cylinder is in its passive period.

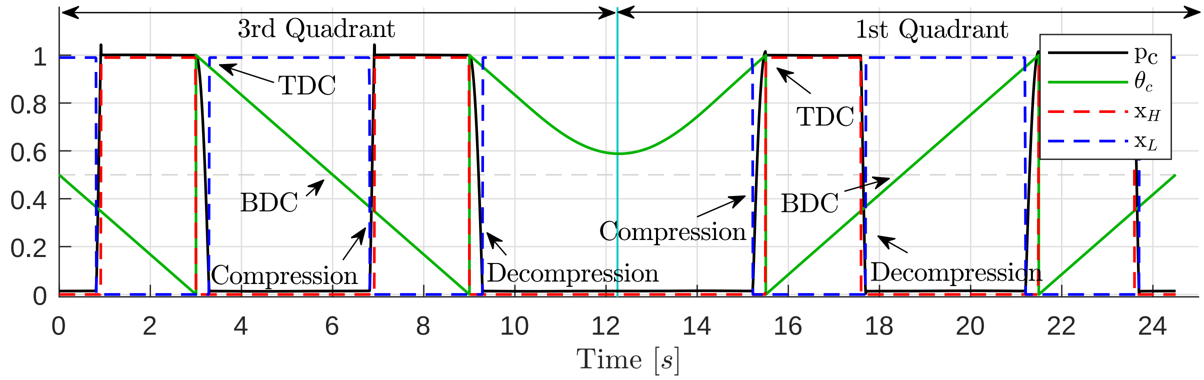


Figure 6.20: Behaviour of cylinder 22

Figure 6.21 shows the flows through the HPV and LPV for the same cylinder. Note that for the first half, where the motor is operating in the third quadrant, the flows are negative. This is because the positive direction is defined when considering the DDM as a motor, where the flow through the HPV goes from the high pressure line p_H to the cylinder chamber p_c and the flow through the LPV from p_c to the low pressure line p_L.

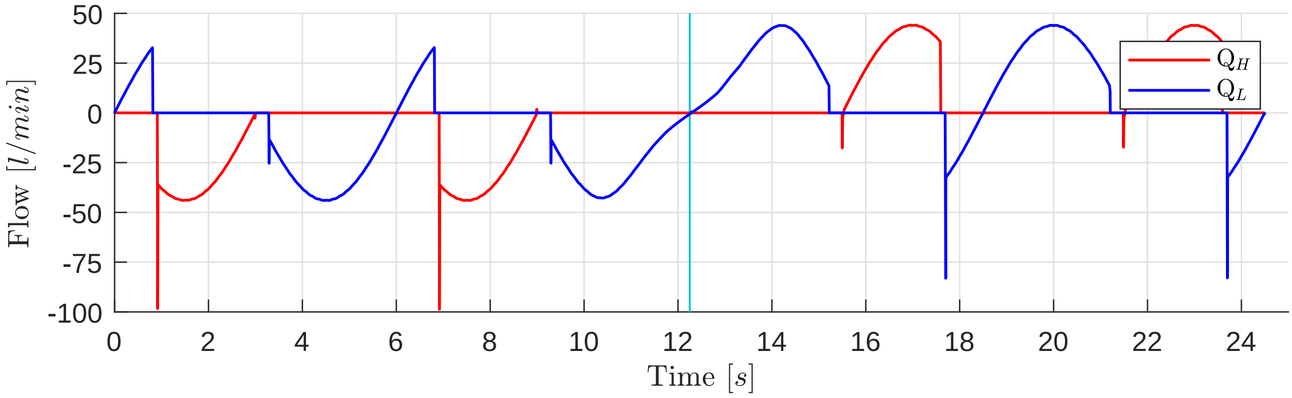


Figure 6.21: Flows over HPV and LPV for cylinder 22

As for the mode switch investigated in section 6.1, the total flows in and out of the DDM are plotted in figure 6.6 in order to check for possible issues related to the mode switch.

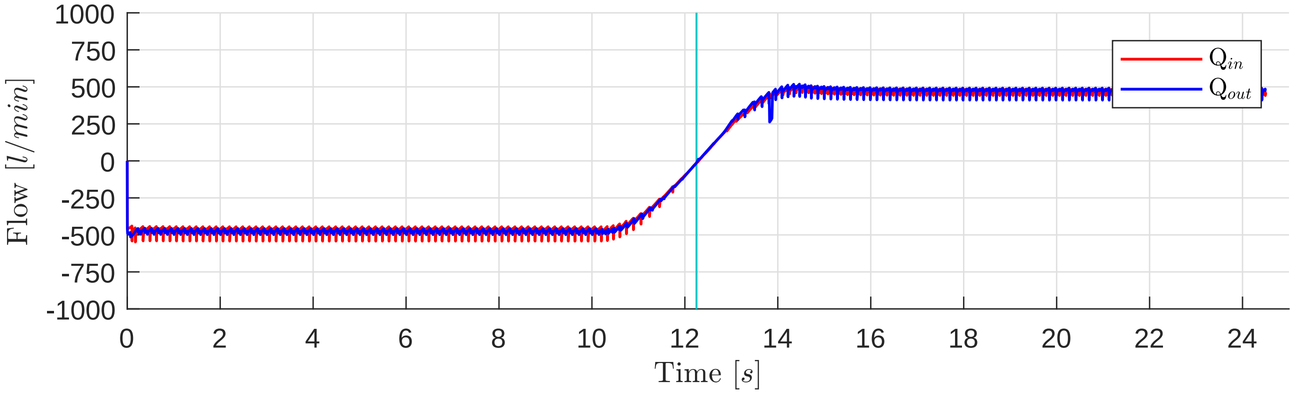


Figure 6.22: Total flow in and out of the motor

One peak is observed at $t \approx 14s$. This cylinder is identified as cylinder 31. The flows through the HPV and LPV of cylinder 31 are plotted in figure 6.23.

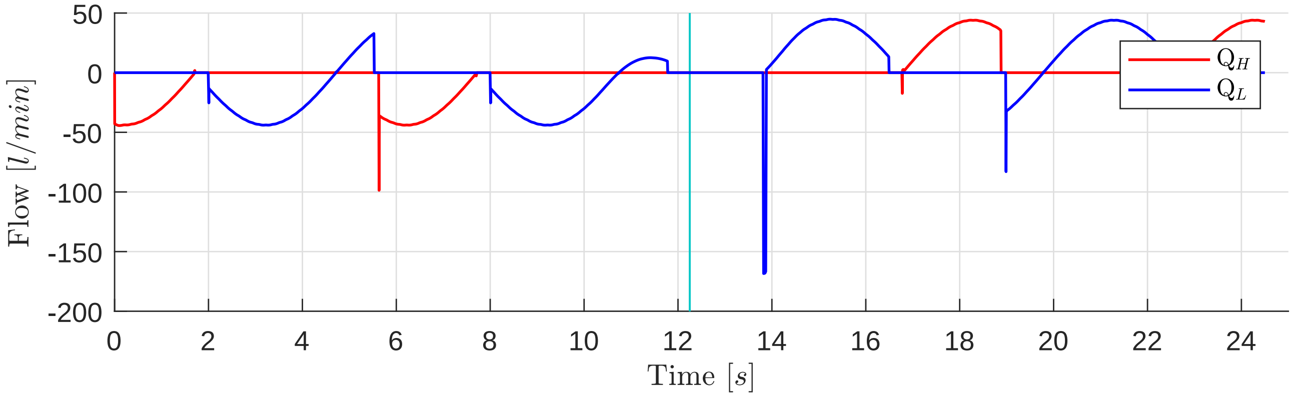


Figure 6.23: Flows over HPV and LPV for cylinder 31

A negative flow peak over the LPV means that there is relatively higher pressure in the cylinder chamber than in the low pressure line. This peak is a result of the behaviour depicted in figure 6.24. The mode is switched while the cylinder is in the compression sequence. As the compression sequence in the 3rd quadrant is a function of the control parameter α, the valve timing fails as the operation mode is switched. The low pressure level between $t \approx 13s$ and $t \approx 14s$ together with the movement of the cylinder piston indicates that the cylinder might experience cavitation in a real-life scenario.

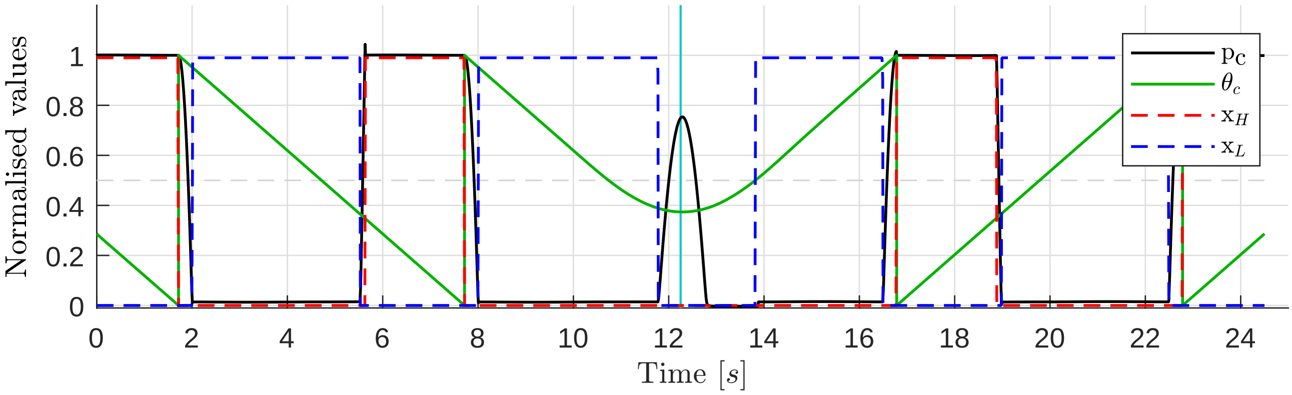


Figure 6.24: Behaviour of cylinder 31

This issue could be countered in different ways, for example by reopening the LPV at the same angle that it closed at if the motor shaft velocity is reversed during the compression sequence. This ensures safe operation of the DDM. Figure 6.25 shows how the cylinder behaves after the new statement is added to the control system.

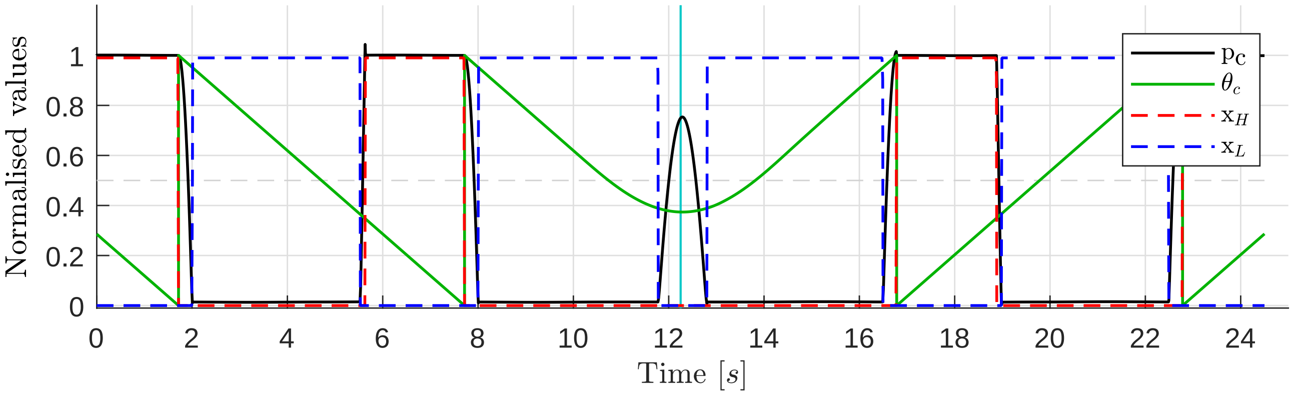


Figure 6.25: Behaviour of cylinder 31, with improved mode switching

Figure 6.26 confirms that the flows are behaving as normal.

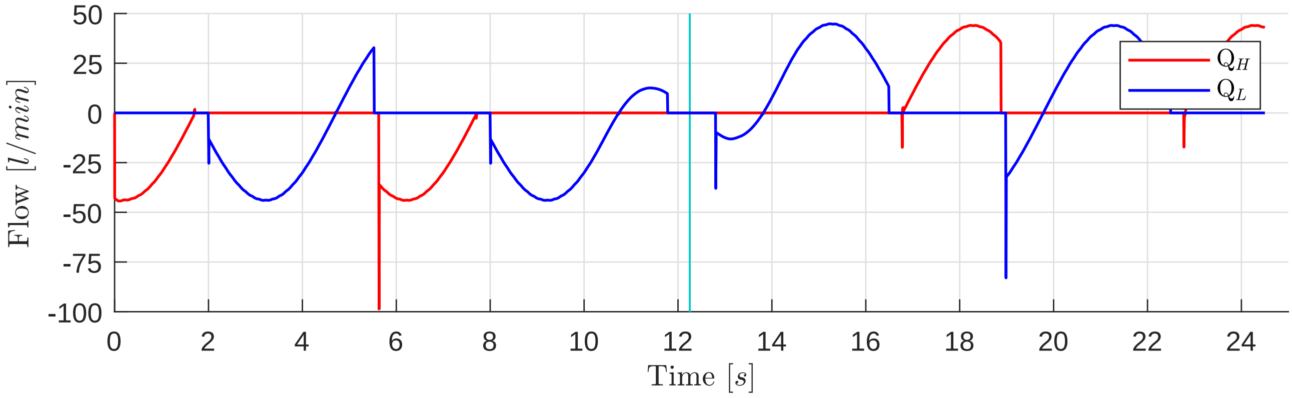


Figure 6.26: Flows over HPV and LPV for cylinder 31, with improved mode switching

Figures 6.27 and 6.28 shows that a mode switch from the third to the first quadrant during the compression of a cylinder does not have a directly negative impact on the behaviour of the cylinder. This is analogous to the decompression of a cylinder during the mode switch between first and third quadrant as depicted in figures 6.11 and 6.11.

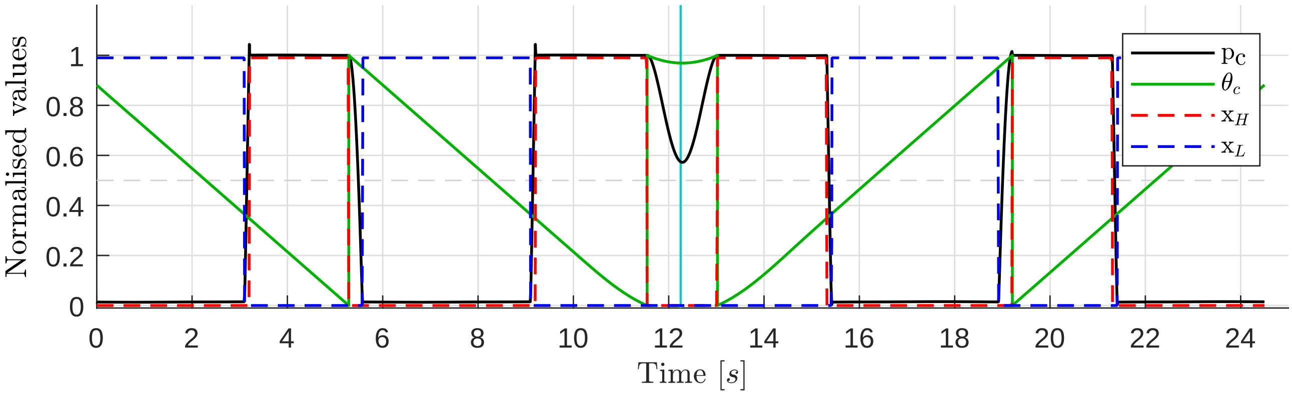


Figure 6.27: Behaviour of cylinder 6, decompression sequence during mode switch

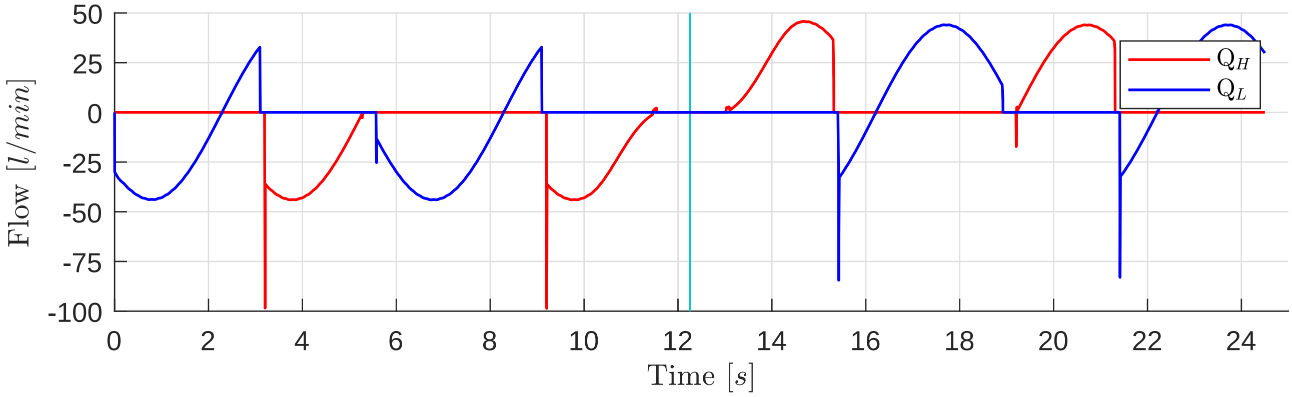


Figure 6.28: Flows over HPV and LPV for cylinder 6

The reference tracking is not particularly affected by the switch improvement in this case, neither is the torque nor the control parameter α. This is because pressure levels of $0 - 5bar$ does not contribute to any considerable torque to the motor shaft.

The total flows in and out of the motor are now plotted again to ensure that the motor is operating as expected:

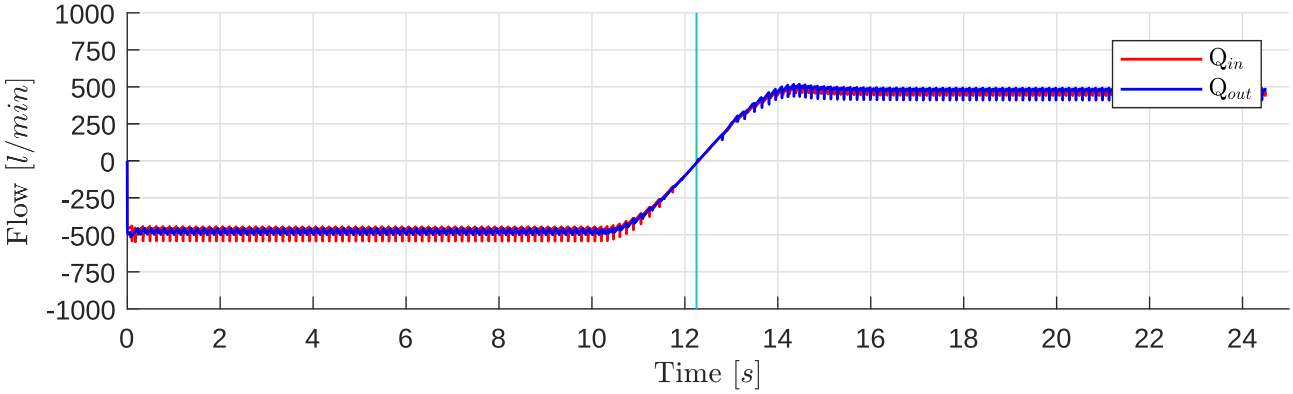


Figure 6.29: Total flow in and out of the motor, with improved mode switching

No peaks are observed, hence no further action is taken.

6.3 Other Mode Switches

This section presents other mode switches that are important to check for a DDM used for a winch application. As discussed in section 3.3.1, operation in the fourth quadrant is not considered a realistic scenario and is therefore not examined further. However, the second quadrant may be briefly used when lowering a light load, also mentioned in section 3.3.1. Thus, four additional mode switches must be examined:

- From 1st to 2nd quadrant
- From 2nd to 3rd quadrant
- From 3rd to 2nd quadrant
- From 2nd to 1st quadrant

The mode switch from the second to the first quadrant is not a realistic scenario for the considered load cases. It implies that the DDM is first working as a motor in CW-direction, with a load torque working in CCW-direction. For the DDM to switch from this scenario (second quadrant) to the first quadrant, the direction of the load and the velocity reference must be reversed. As argued when introducing the valve timing strategy for the second quadrant in section 3.3.1, operation in the second quadrant is only expected if a light payload struggles to follow a negative velocity reference. Hence, as long as the velocity reference is ramped when changing direction, the high inertia in the system is expected to ensure a transition through the third quadrant (CW pump mode). This mode switch scenario is therefore not investigated further.

6.3.1 First to Second Quadrant

In contrast to the mode switch from the second to the first quadrant, a mode switch from the first to the second quadrant is a possible scenario, although not considered to occur frequently. If the winch is hoisting a light load when the reference velocity changes direction, the gravitational forces of the payload may not be high enough to overcome the friction and the inertia of the system.

To force the control system to switch from the first to the second quadrant, the applied load torque and the velocity reference is changed from a positive value to a negative value over a given time period. Figure 6.30 plots the velocity reference and the motor shaft velocity, and figure 6.32 plots the manipulated load torque and the motor torque.

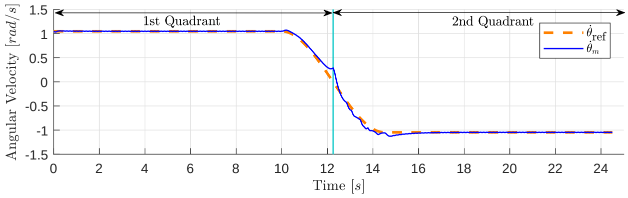


Figure 6.30: Shaft velocity and reference

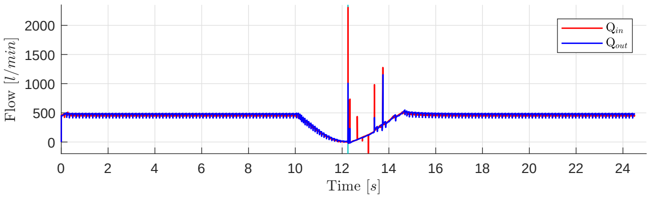


Figure 6.31: Total flow in and out of the motor

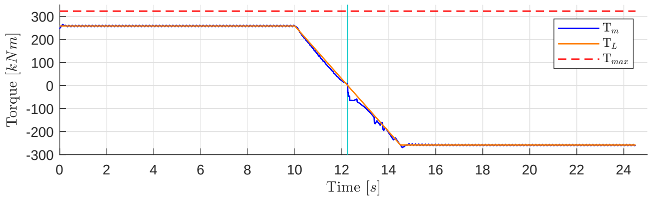


Figure 6.32: Motor torque and load torque

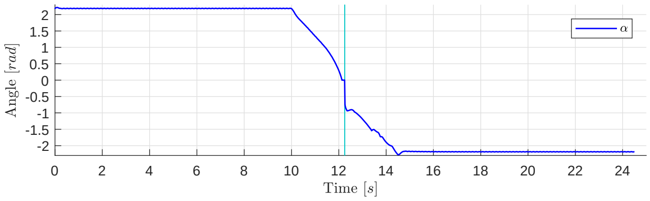


Figure 6.33: Control parameter α

The control parameter α is plotted in figure 6.33. A sudden drop in the value of α is observed at $t \approx 12s$ as the motor switches quadrant. Right before the mode switch, the value is flat zero over a small interval. This is because the value of α is forced to not go under zero to ensure operation in the first quadrant prior to the change in direction of the velocity set-point.

As the motor switches operation mode, the active period of the cylinders is flipped along the TDC-BDC axis, as can be seen by studying the valve timing diagrams for quadrants 1 and 2 (figures 3.16 and 3.18). This means that cylinders that were in the passive period when the motor was operating in the first quadrant, now may be activated without the ability to undergo the correct pressurisation sequence when the control system switches mode and the motor shaft starts to rotate in the CW-direction. This is the reason for most of the flow peaks observed in figure 6.31. Figures 6.34 and 6.35 shows the behaviour of cylinder 3 which reacts as described.

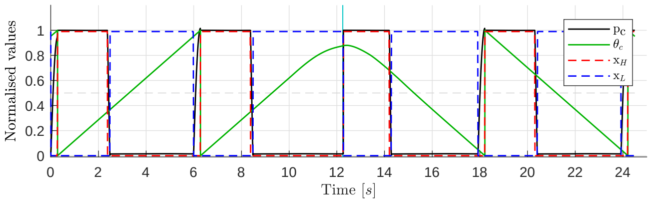


Figure 6.34: Behaviour of cylinder 3

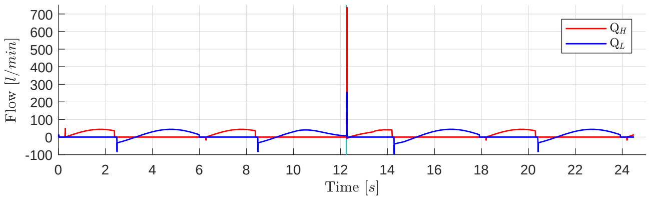


Figure 6.35: Flows through the HPV and LPV of cylinder 3

However, some of the flow peaks are because the mode switch is done when a cylinder is decompressing, producing a pressure build-up similar to the one described in section 6.1, figures 6.8 and 6.9. This issue is solved in the same manner, by allowing the cylinder to reopen the HPV at the same angle that it closed as the motor shaft velocity changes direction. Figure 6.36 shows the total flows in and out of the motor after the implemented correction. The remaining flow peaks are due to the reactivation of the cylinders as described earlier.

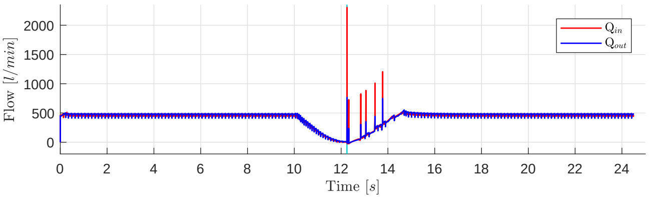


Figure 6.36: Total flow in and out of the motor

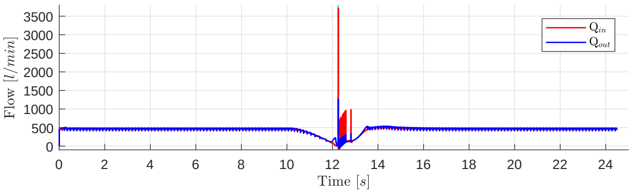


Figure 6.37: Total flow in and out of the motor, fast load switch

Figure 6.37 shows how the peaks are more consentrated if the change in load direction is done fast. The resulting peak at the moment where the mode is switched is much higher as a consequence.

It should be noted that this mode switch is provoked by manipulating the load torque T_L. If the DDM is actuating a winch with a light payload as discussed, the load torque will not change drastically as shown here. Thus, the active range of the control parameter α will be reduced, with less flow peaks as a result.

6.3.2 Third to Second Quadrant

If the motor is running in CW-pump mode (third quadrant) with a light load and struggles to follow a clockwise velocity reference, the control system might decide to change operation mode to the second quadrant to increase the payload's downwards velocity. Therefore, the mode switch between these two quadrants is evaluated in case this situation occurs.

A study of the valve diagrams of the second and third quadrant (figures 3.18 and 3.20) shows that the active period of the cylinders is flipped along the TDC-BDC axis, but the velocity direction is the same for the two quadrants. As discussed in 3.3.1, a decreasing α in CW-pump mode (third quadrant) means that the deactivated cylinders are de-pressurised by simultaneoulsy opening the LPV and closing the HPV, thus leading to flow peaks. For the CW-motor mode (second quadrant), an increasing α reactivates available cylinders by simultaneously opening the HPV and closing the LPV. Thus, a mode switch from the third to the second quadrant gives many flow peaks, as seen in figure 6.39. Figures 6.38 and 6.40 shows the velocity and torque behaviour generated to provoke the mode switch, and figure 6.41 shows the corresponding control parameter α.

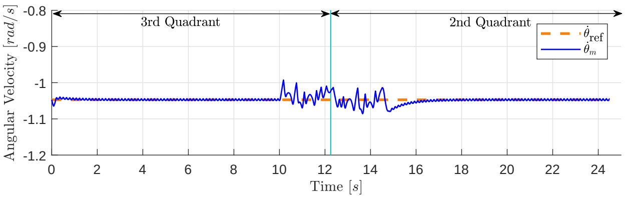


Figure 6.38: Shaft velocity and reference

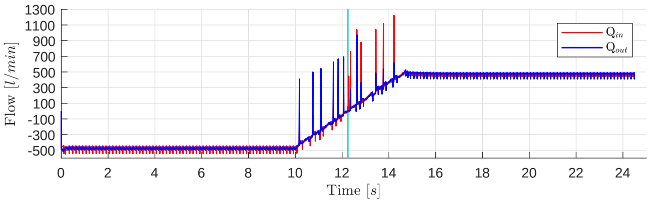


Figure 6.39: Total flow in and out of the motor

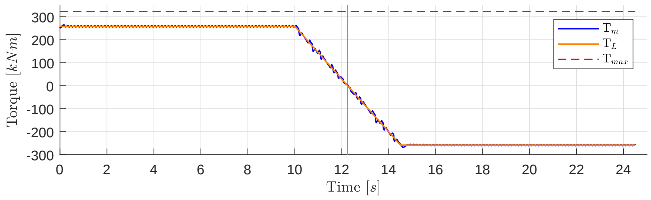


Figure 6.40: Motor torque and load torque

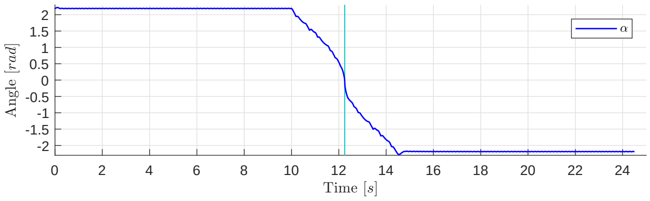


Figure 6.41: Control parameter α

Figures 6.42 and 6.43 shows how a cylinder is affected when the control parameter α is decreasing at a rate close to that of the motor shaft velocity. The cylinder is activated and deactivated repeatedly, causing the valves to open against a high pressure difference.

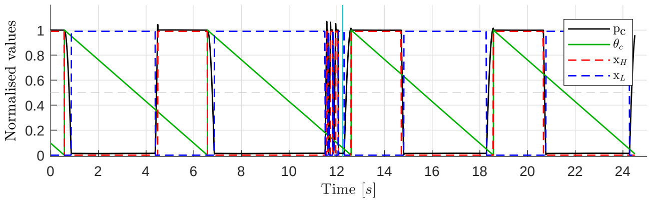


Figure 6.42: Behaviour of cylinder 39

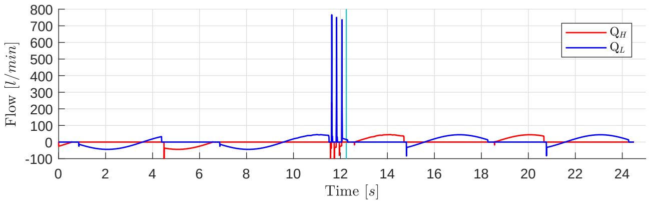


Figure 6.43: Flows through HPV and LPV of cylinder 39

As the flow peaks related to this mode switch are solely determined by the possibility of the cylinders to reactivate/deavtivate, no action is taken to counter this.

6.3.3 Second to Third Quadrant

As discussed in the beginning of this chapter, it is expected that if the motor first enters the second quadrant, it will transition through the third before going to any other quadrants. This mode switch must therefore be evaluated in case it occurs.

In contrast to the mode switch between the third and the second quadrant, the changes in the value of α does not trigger abrupt deactivation/reactivation of the motor cylinders. As described in section 3.3.1, a decrease in α when the motor operates in the second quadrant initiates the depressurisation sequence of the cylinders. Correspondingly, an increase in α when the motor operates in the third quadrant initiates the pressurisation sequence of the cylinders. Hence, after correcting for some special cases of decompression during the mode switch, no flow peaks are observed for this situation, as shown in figure 6.45. Figures 6.44 and 6.46 shows the velocity and torque behaviour generated to provoke the mode switch, and figure 6.47 shows the corresponding control parameter α.

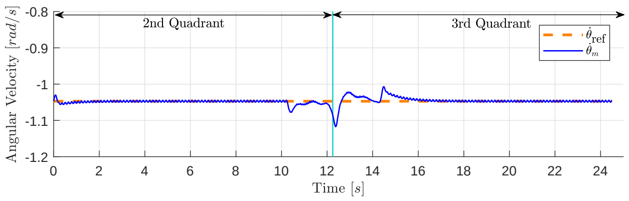


Figure 6.44: Shaft velocity and reference

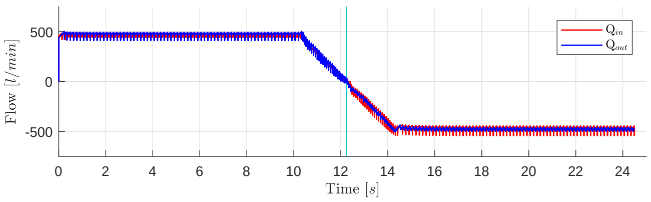


Figure 6.45: Total flow in and out of the motor

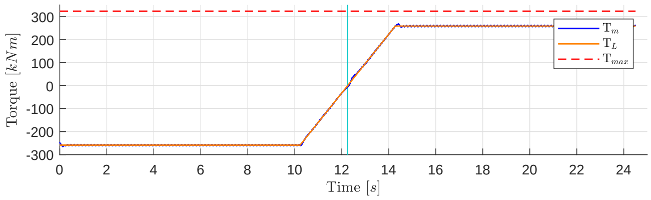


Figure 6.46: Motor torque and load torque

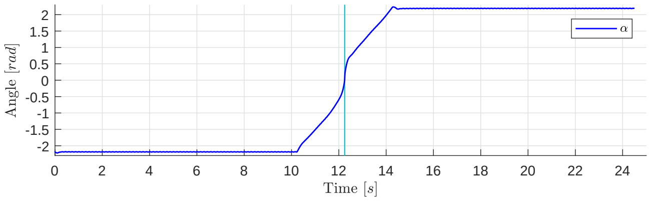


Figure 6.47: Control parameter α

7 | Winch System Results

The considered winch system from chapter 4 is connected to the digital displacement motor, to form a complete winch model. This chapter presents the results from simulations of this model. Tested and improved valve timing algorithms from chapter 6 is implemented in order to switch between motor operation modes correctly, during simulation of the winch system. As described in the problem statement (section 1.2), the winch system must be able to handle both heavy and light loads, while maintaining stable operation. The motor is given velocity references during simulations, in order to hoist or lower the payload. The behaviour of the heavy or lightly loaded system it then observed. Finally, the motor is given a sinusoidal velocity reference, in order to simulate traditional heave compensation mode. In order to control the motor torque, the displacement parameter α is controlled by the use of a PID-controller with feed forward signal, as described in section 3.3. Controller parameters for all hoisting and lowering operations are collected in table 7.1, while the controller paramters used for heave compensation is collected in table 7.2.

Parameter	**Value**
K_p	1.7
T_i	0.85
T_d	0

Table 7.1: Hoisting and lowering PID Parameters

Parameter	**Value**
K_p	1.7
T_i	0.7
T_d	0

Table 7.2: Heave compensation PID parameters

7.1 Payload Hoisting

This section presents simulation results from hoisting of light and heavy payloads. Hoisting is achieved by giving the motor a positive angular velocity reference in form of a ramp, starting and ending at zero velocity. Initially, the payload is at rest in the selected operation point, $z_{pl} = -200m$.

7.1.1 Heavy Load

A payload with a mass of $m_{pl} = 20\ 000kg$ is connected to the winch, which is the maximum capacity of the considered hoisting equipment. The angular velocity reference, $\dot{\theta}_{ref}$, can be observed in figure 7.1, in addition to the actual motor shaft angular velocity, $\dot{\theta}_m$. Initially, the motor speed oscillates when given a velocity reference of zero. The effect is decreasing as the reference velocity is ramped up. This slightly degrades the reference tracking in the interval $t = [6, 8]s$. A slight overshoot is observed at $t = 11s$ before the system follows within acceptable margins up until $t = 21s$, where the same oscillations around zero are observed. During the entire sequence, the DDM operates in the first quadrant (CCW motor mode).

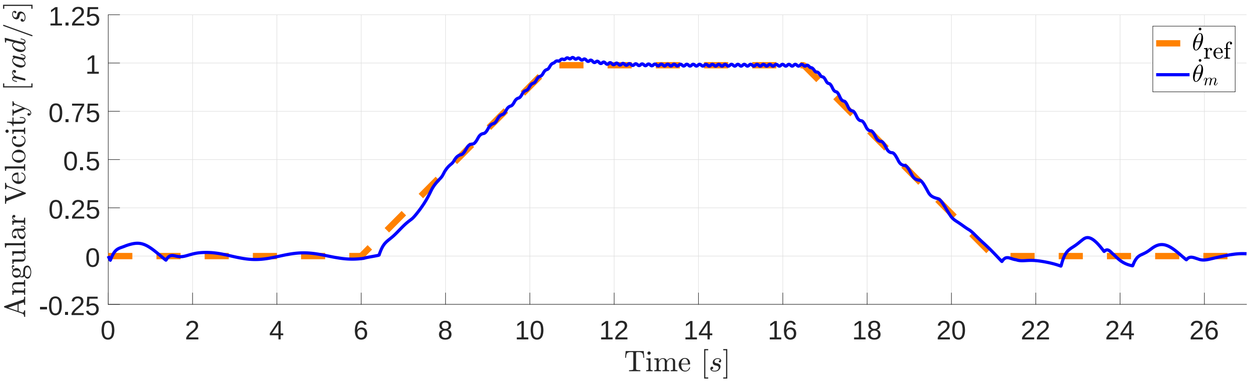


Figure 7.1: Angular velocities during hoisting - oscillations around zero

Figure 7.2 shows the payload position as a consequence of the above motoring sequence, starting at an initial position of $z_{pl} = -200m$. Ideally, the payload position is constant when the motor is given a velocity reference of zero. However, the payload position is affected by the mentioned oscillations around zero motor speed, translating into position oscillations of maximum $7.68cm$ at these points.

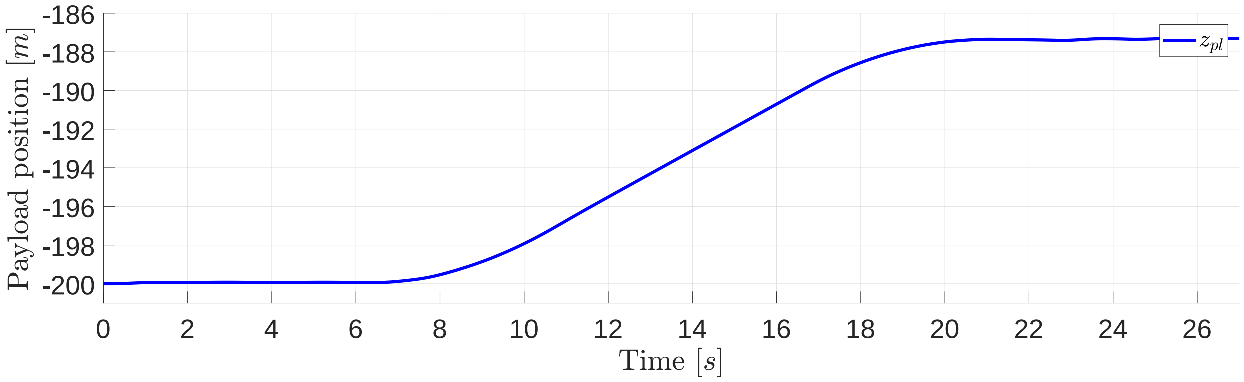


Figure 7.2: Heavy load hoisting sequence

Figure 7.3 presents the displacement parameter α during the motoring sequence. As mentioned, the value of this parameter is controlled by the PID-controller with feed forward signal.

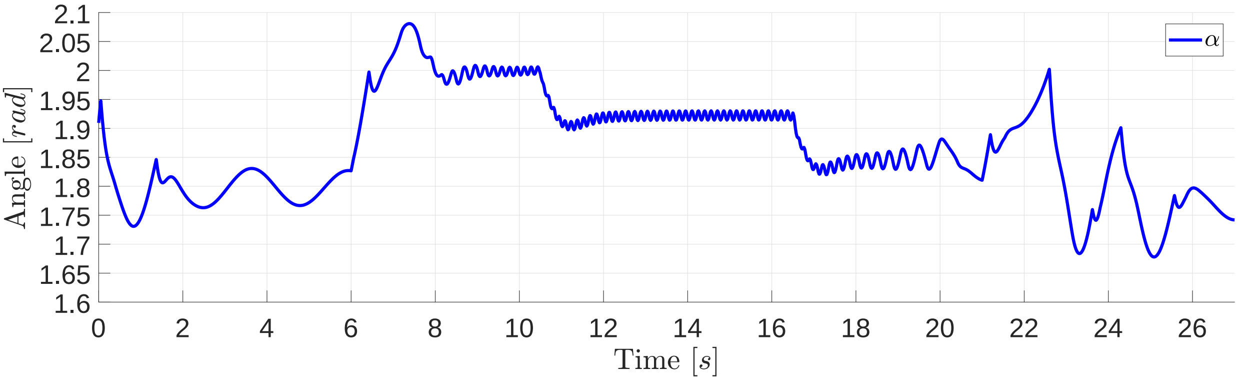


Figure 7.3: Displacement parameter α during hoisting of the heavy load

Figure 7.4 presents the motor- and load torque, T_m and T_L, during the motoring sequence. Due to the elastic steel wire connected to the winch drum, the load torque is dynamic. As can be seen, the motor torque level follows the value of α from figure 7.3.

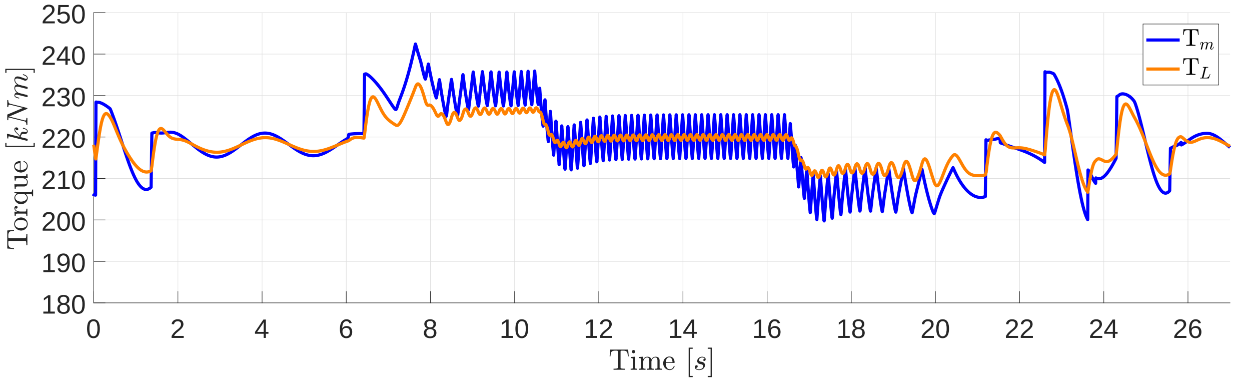


Figure 7.4: Torques during hoisting of the heavy payload

As part of the system validation process, the oil flows into and out of the motor, Q_{in} and Q_{out}, are examined. Figure 7.5 shows the motor oil flows, with observable peaks located at regions where the motor shaft speed oscillates.

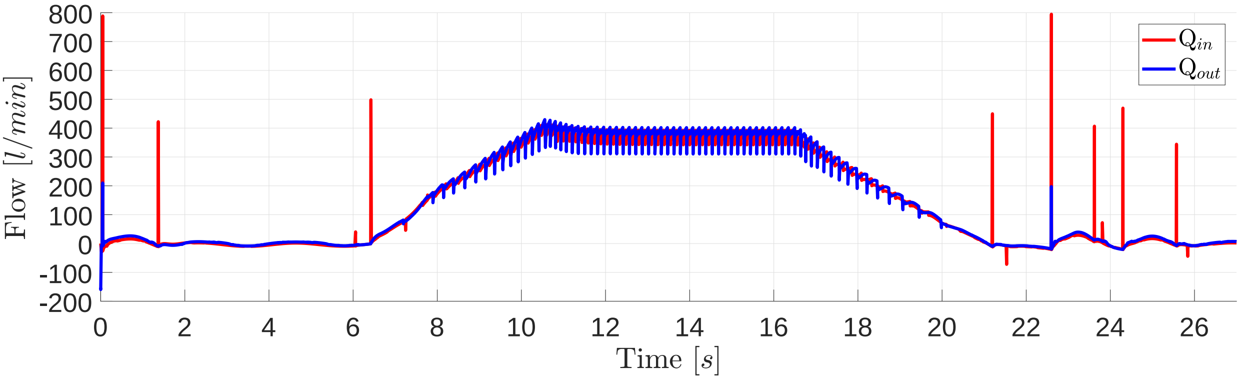


Figure 7.5: Motor flows during hoisting of the heavy load

According to the results presented above, the system is functioning correctly, and capable to handle the heavy payload within acceptable margins. However, there is potential for improvement in the region where the motor tries to follow a velocity reference of zero. Therefore, a method for locking the motor shaft in these regions is suggested, here called *lock mode*. By implementing this, the motor and payload will stay stationary at $\dot{\theta}_{ref} = 0 rad/s$. Also, the flow peaks associated with zero velocity reference will be eliminated. Lock mode is achieved by closing the valves at a point of low velocity error, consequently locking the motor shaft. Lock mode is therefore introduced to the system, and the heavy payload hoisting sequence repeated. Figure 7.6 shows the repeated motoring sequence, with lock mode activated. As can be seen, the use of lock mode keeps the motor shaft stationary in the desired periods. Also, the reference tracking is improved when comparing to figure 7.1.

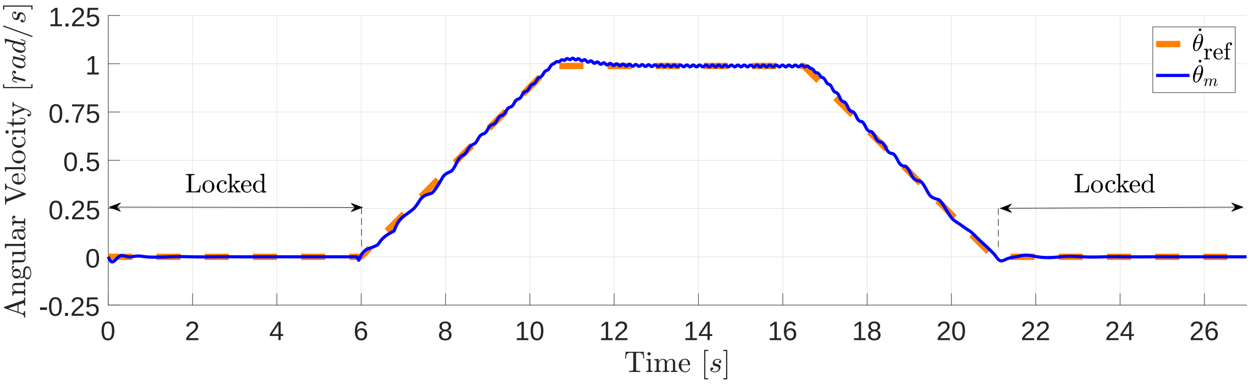


Figure 7.6: Angular velocities - introduced lock mode

As a consequence, the payload remains almost stationary while in lock mode, with a maximum deviation of $0.4mm$. When exiting lock mode, the payload position does not drop when the reference velocity is ramped up, in contrast to the results presented in figure 7.2. This is indicated in figure 7.7 at $t \approx 6s$.

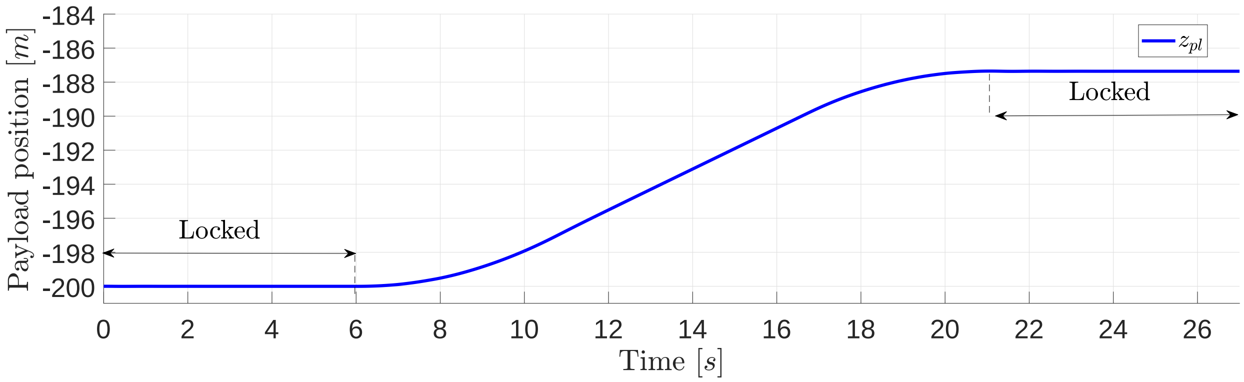


Figure 7.7: Heavy load hoisting sequence with enabled motor lock mode

Figure 7.8 shows the displacement parameter α when lock mode is enabled. It should be noted that during lock mode, the motor does not respond to the changes in α as all valves are kept closed.

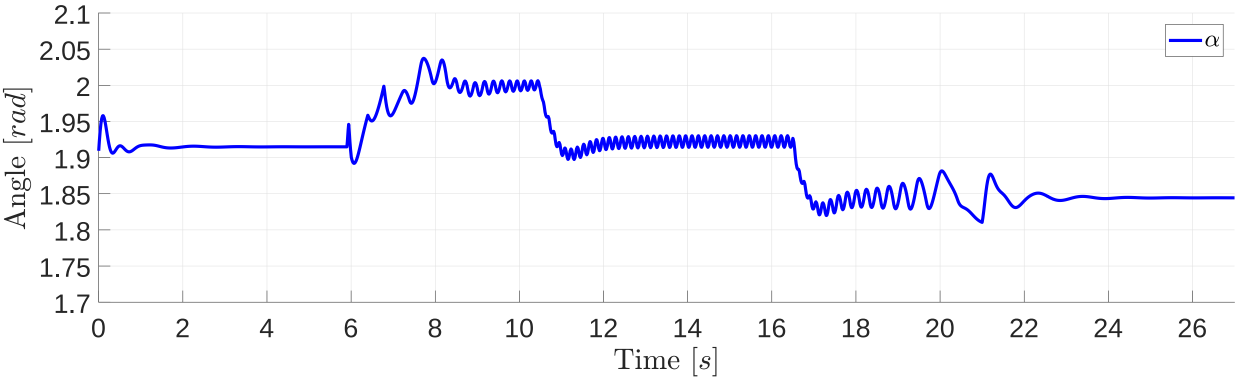


Figure 7.8: Displacement parameter α with enabled lock mode

The motor torque follows the updated values of α, seen in figure 7.9.

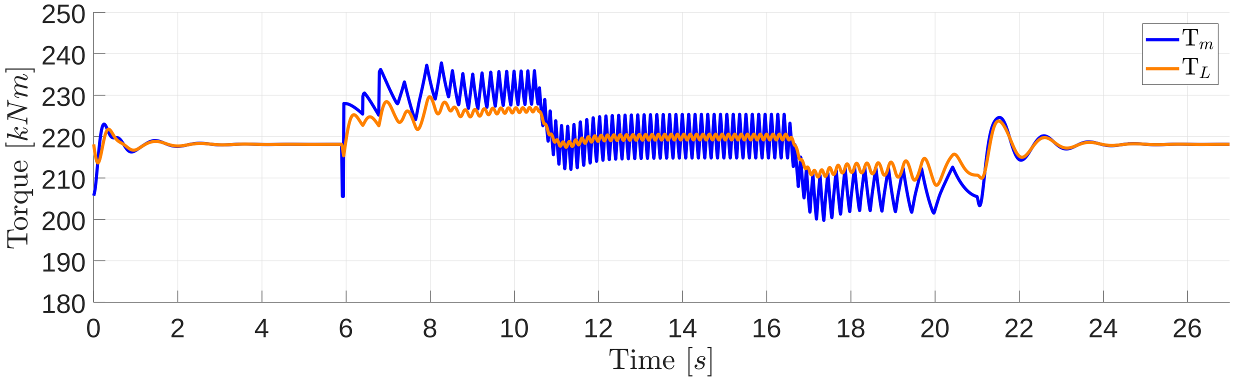


Figure 7.9: Torques with lock mode enabled

Due to lock mode, the flow peaks related to the oscillations around zero velocity are eliminated, seen in figure 7.10. However, when applying lock mode, the entrapped oil volume inside the cylinders are either slightly compressed or decompressed. This is due to the force applied to the pistons by the load torque. Therefore, when exiting lock mode, the valves are opened against a pressure difference. This produces the flow peaks seen in figure ref at $t \approx 6s$. The remaining flow peaks, located at $t \approx 7s$, is due to reactivated cylinders as described in section 3.1.2.

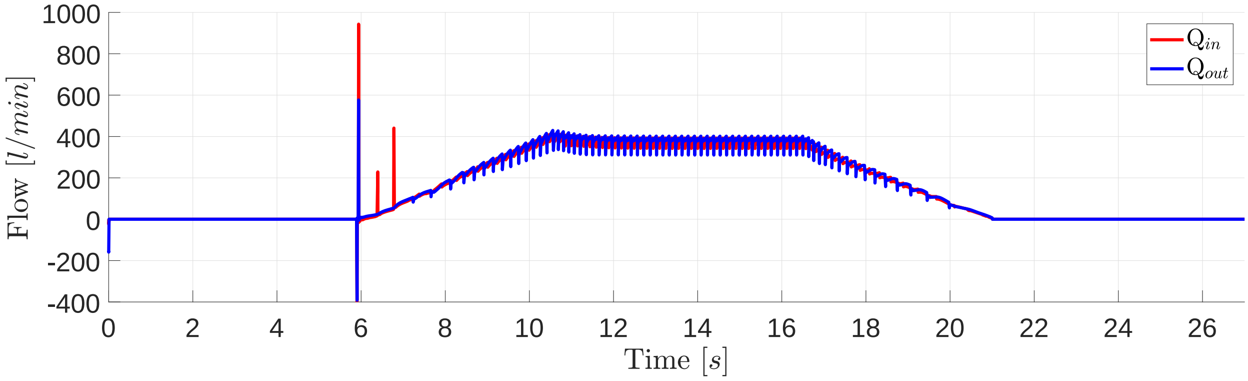


Figure 7.10: Motor flows during the hoisting sequence and lock mode

Due to the benefits from using lock mode, the following results in this chapter will be presented with this function enabled. It is worth to emphasize that lock mode is not reactivated before the system reaches a certain motor shaft velocity tolerance and the velocity reference is zero, in order to close the valves at favourable conditions.

7.1.2 Light Load

This section demonstrates hoisting of a light load, and system behaviour during the sequence. In this case, the payload is disconnected from the winch. Thus, the only load applied to the winch is the gravitational load generated by the suspended wire mass. The initial position of $z_{pl} = -200m$ translates into $1132kg$ wire mass. Figure 7.11 presents the angular velocity reference in addition to the actual motor speed. The DDM is operating in the first quadrant during entire sequence (CCW motor mode). As a reminder, lock mode is enabled.

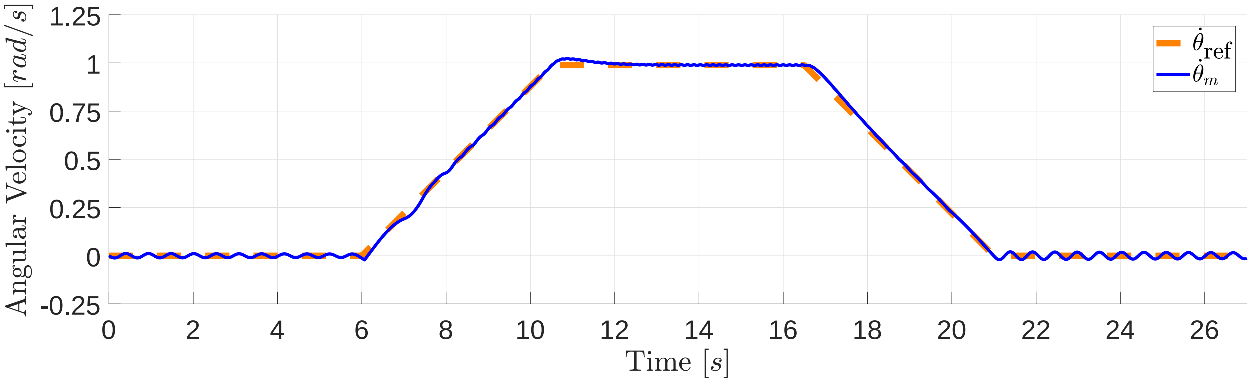


Figure 7.11: Angular velocities during hoisting

Figure 7.12 presents the position of the hook that conventionally connects to the payload, showing normal behaviour.

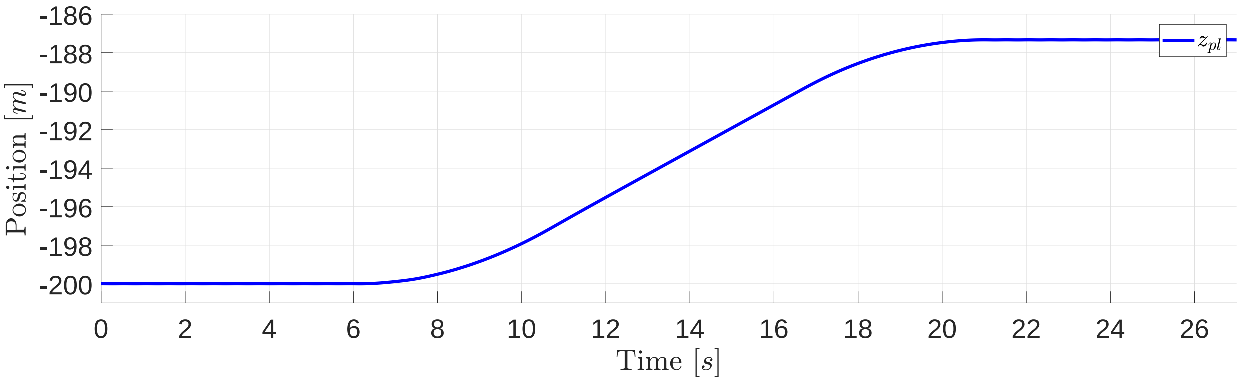


Figure 7.12: Light load hoisting sequence

During hoisting of the light load, not much displacement is needed, and the control system applies a low value of α to produce the necessary torque, seen in figure 7.13.

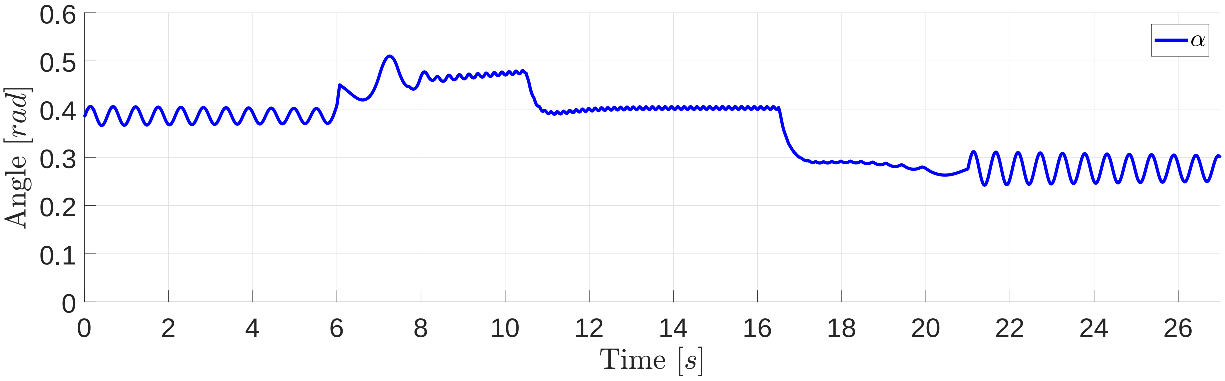


Figure 7.13: Displacement parameter α during light load hoisting

Figure 7.14 shows the motor- and load torque during the hoisting sequence. The initial value of T_L corresponds to the load torque generated by the wire weight. At $t = 6s$, the motor torque must be raised in order to accelerate the inertia of the winch system. Analogously, the control system reduces the motor torque to decelerate at $t \approx 16.5s$.

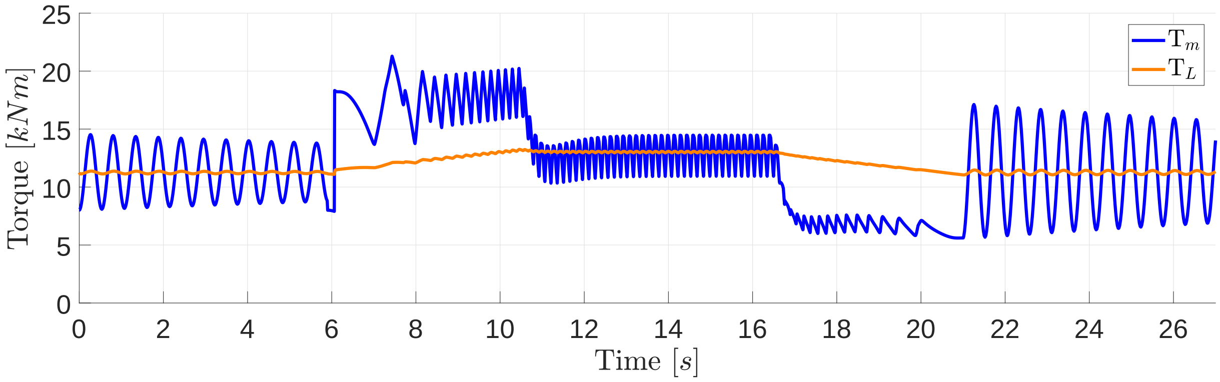


Figure 7.14: Torques during hoisting of the light payload

It should be noted that the oscillating motor torque during lock mode is due to the lightly loaded system that can give oscillations on the motor shaft. The resultant oil flow in and out of the motor is presented in figure 7.15. Analogously to figure 7.10 in the previous section 7.1.1, the flow peak observed at $t \approx 6s$ is due to reactivation of one motor cylinder.

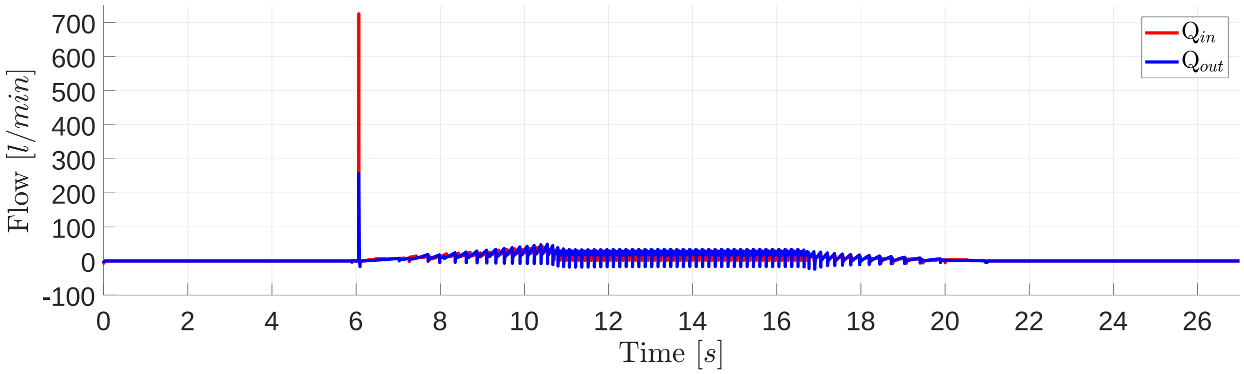


Figure 7.15: Motor flows during hoisting of the light payload

7.2 Payload Lowering

This section presents simulation results from lowering of light and heavy payloads. Lowering is achieved by giving the motor a negative angular velocity reference in form of a ramp, starting and ending at zero velocity. Initially, the payload is at rest in the selected operation point, $z_{pl} = -200m$.

7.2.1 Heavy Load

System behaviour and results from lowering of the 20 000kg payload is covered in this section. Due to the negative angular velocity reference presented in figure 7.16, the motor will operate in pump mode during the lowering session. This can be observed at $t = 6s$, where the motor switches from first quadrant in lock mode to third quadrant operation (CW pumping). The system re-enters lock mode after completion of the lowering sequence. The vertical blue lines indicate switching between motor modes.

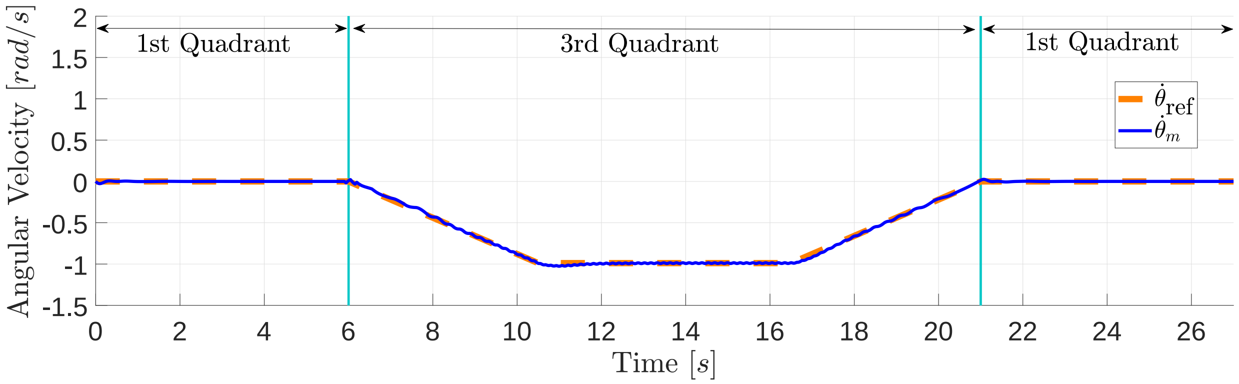


Figure 7.16: Angular velocities during lowering

Lowering of the heavy payload is presented in figure 7.17, showing normal behaviour.

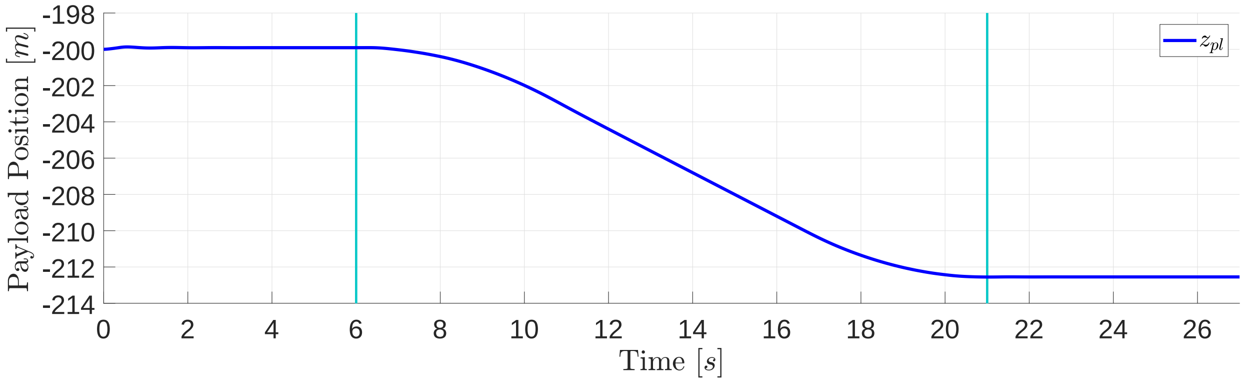


Figure 7.17: Heavy payload lowering sequence

Since the motor operates as a pump during the lowering sequence, the displacement parameter α is manipulated by the control system in order to produce a resistance torque T_m resulting in a controlled lowering of the payload. This can be seen in figure 7.18. In order to accelerate the payload downwards, the value of α is reduced at $t = 6s$. Thus, less resistance torque is provided by the pump, resulting in downwards payload movement. Oil is now pumped from the low-pressure port to the high-pressure port of the pump. Constant shaft speed is achieved when the control system increases the value of α at $t \approx 10.5s$. In order to decelerate the payload at $t \approx 16.5s$, the displacement parameter α is further increased by the control system, raising the resistance torque provided by the pump. The torque plot found in figure 7.19 can be used for visual reference.

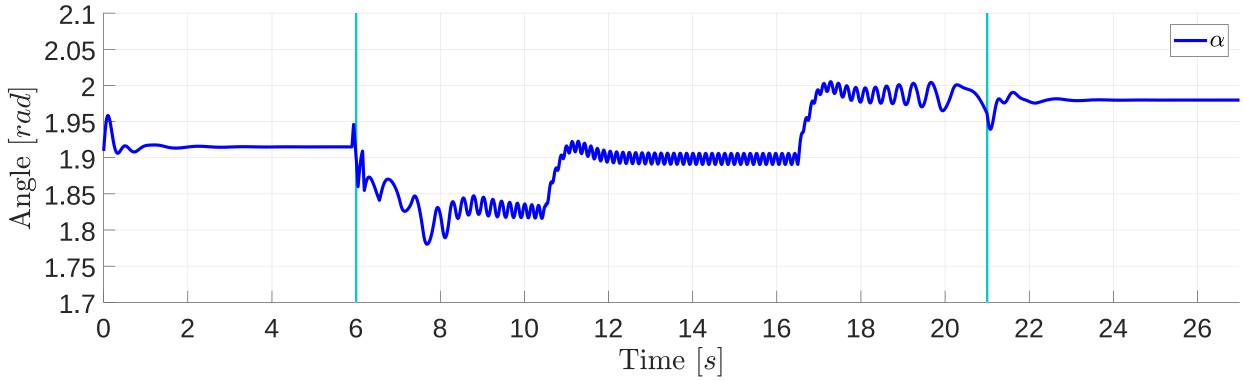


Figure 7.18: Displacement parameter α during heavy payload lowering

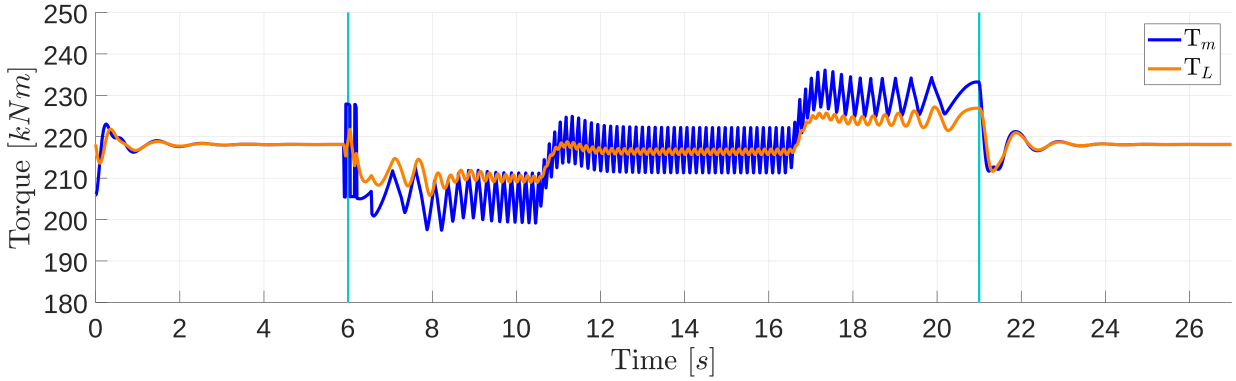


Figure 7.19: Torques during lowering of the heavy payload

The resultant oil flow in and out of the pump is plotted in figure 7.20. As the pump exits lock mode at $t \approx 6s$, a small number of flow peaks are observed. This is due the control system that computes a slightly noisy displacement signal α at this point, seen in figure 7.18. Thus, if the calculated α signal was smoother at this particular point, the flow peaks would be eliminated.

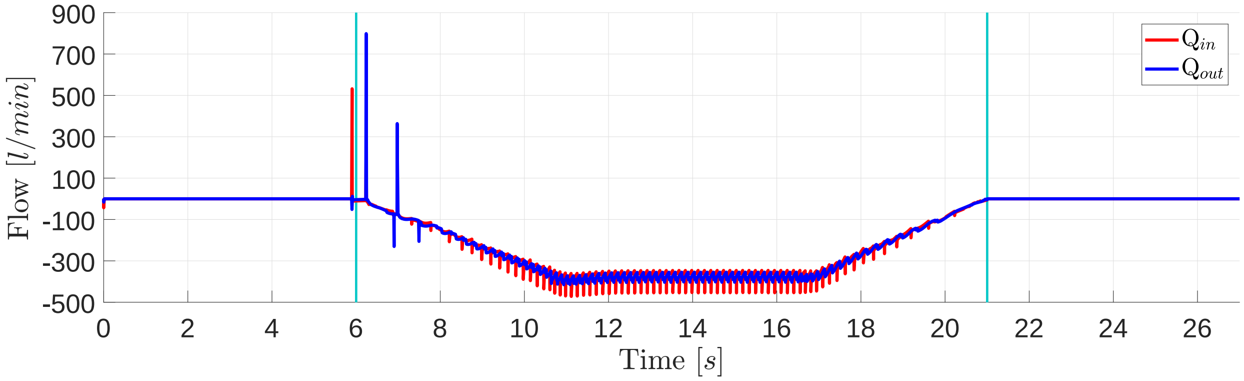


Figure 7.20: Motor flows during lowering of the heavy payload

7.2.2 Light Load

This section presents the results from lowering of a light payload. Similar to section 7.1.2, the only load applied to the winch system is generated by the suspended wire mass. As mentioned in section 3.3.1, the motor is expected to enter the second quadrant (CW motoring) to assist reference tracking when lowering a light load. This is due to the load not being sufficiently high to accelerate the winch inertia fast enough according to the given velocity reference.

Figure 7.21 presents the angular velocity reference and actual motor speed during the lowering sequence. When exiting lock mode at $t \approx 6s$, the system tries to follow the reference by entering the third quadrant, which is normal clockwise pumping. However, when the velocity error becomes too large, the motor enters the second quadrant (CW motoring) at $t \approx 6.5s$ in order to assist reference tracking. When the velocity error is compensated for, the motor re-enters normal clockwise pumping during the rest of the lowering sequence.

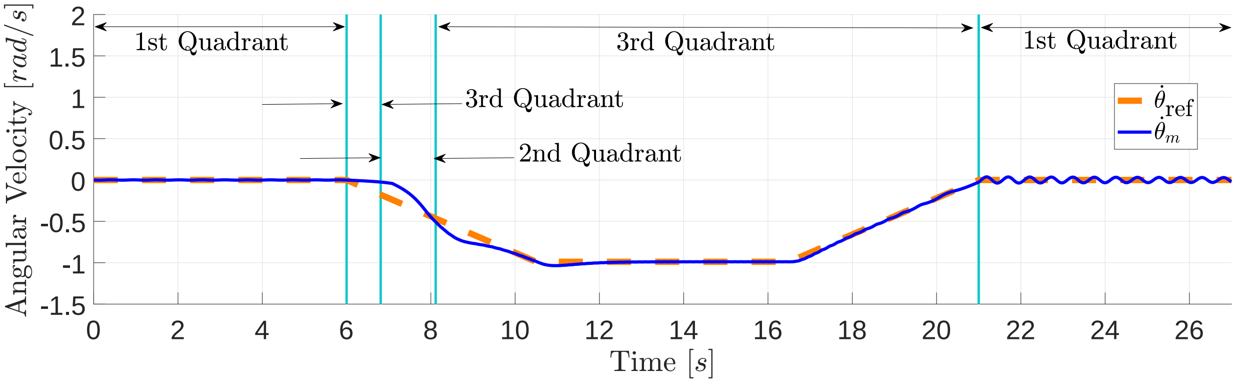


Figure 7.21: Angular velocities during lowering

Figure 7.22 shows the position of the hook that conventionally connects to the payload, confirming normal behaviour.

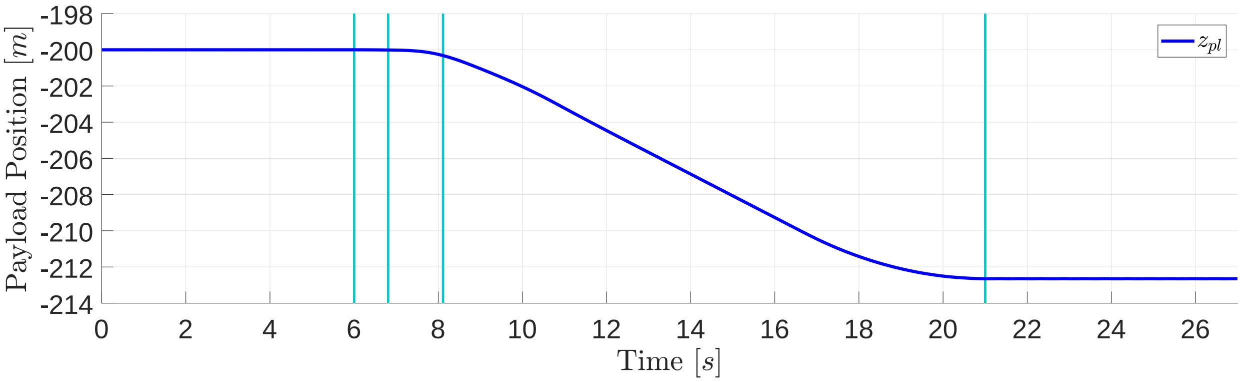


Figure 7.22: Lowering of the unloaded hook

Figure 7.23 shows the displacement parameter α during the lowering sequence. As can be seen, the value is negative in the time period where the motor is in the second quadrant. This means motoring in the clockwise direction, which also can be seen in the torque plot presented in figure 7.24.

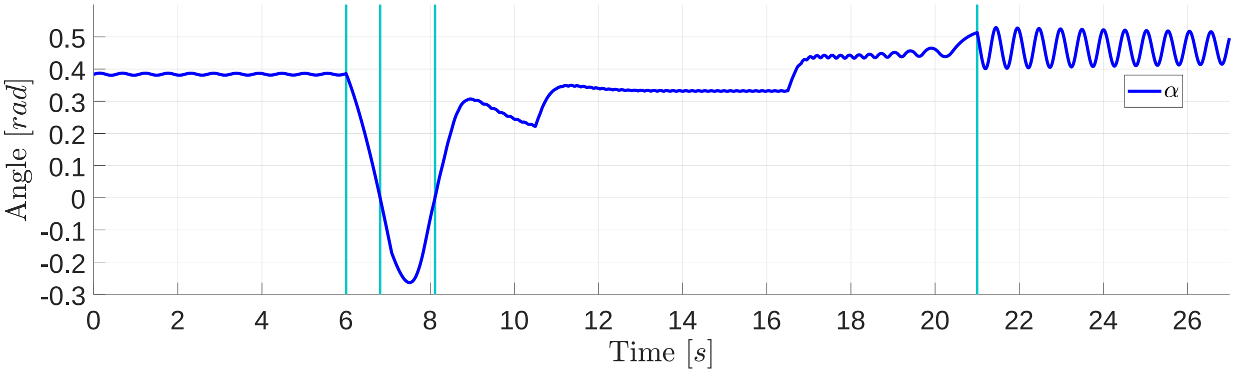


Figure 7.23: Displacement parameter α during light load lowering

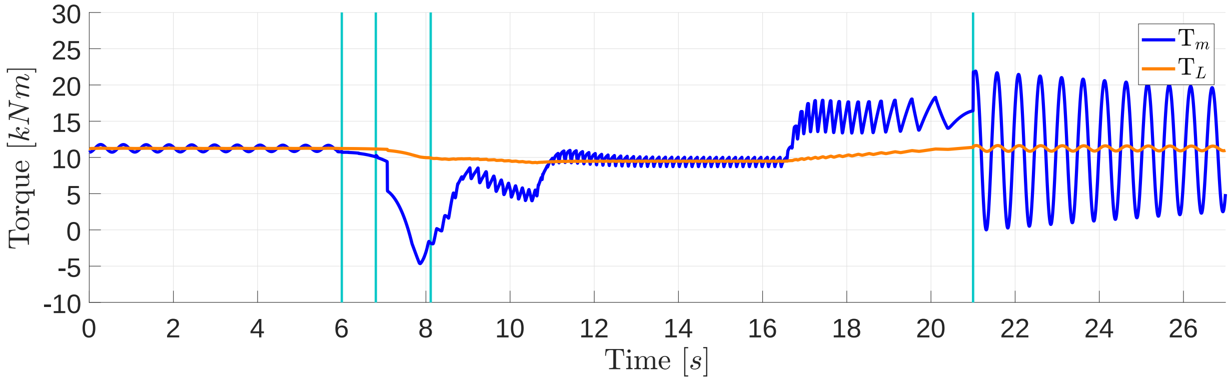


Figure 7.24: Torques during lowering of the light load

Figure 7.25 shows the resultant oil flows into and out of the motor during lowering. A small number of flow peaks can be observed at $t \approx 7s$, where the motor switches from third to the second quadrant. Problems related to this particular mode switching are discussed in section 6.3.2, where it is explained why flow peaks will occur.

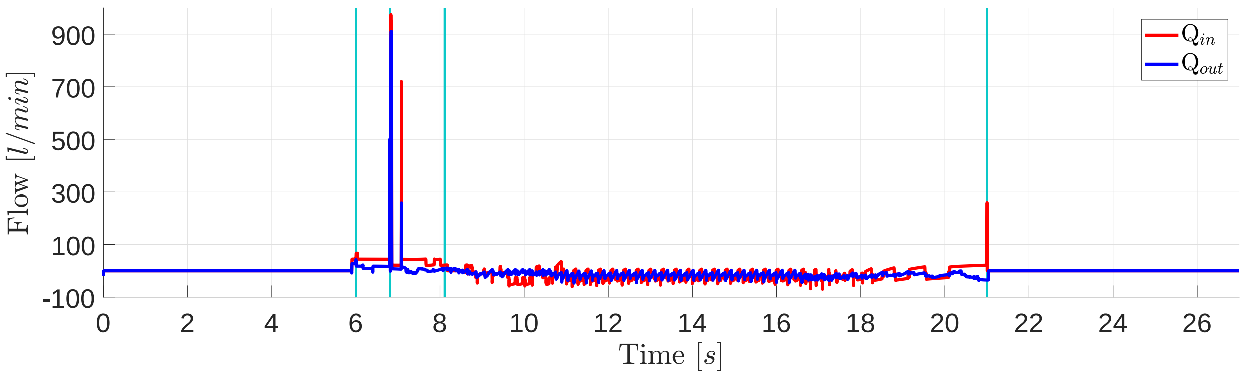


Figure 7.25: Motor flows during lowering of the light load

7.3 Heave Compensation

This section presents results from the system performing traditional heave compensation. Therefore, the motor is given a typical sinusoidal velocity reference signal used for active heave compensation scenarios. During heave compensation, the motor mode will shift between first and third quadrant (CCW motoring and CW pumping). The system is loaded to maximum winch drum capacity, i.e. 20 $000kg$ during this sequence. Figure 7.26 presents the angular velocity reference applied to the motor, in addition to the actual motor speed during heave compensation. Initially, the system operates in the third quadrant, performing clockwise pumping to compensate for upward wave movement. Once given a positive velocity reference at $t \approx 2.2s$, the motor enters the first quadrant, performing counter clockwise motoring in order to compensate for downwards wave movement.

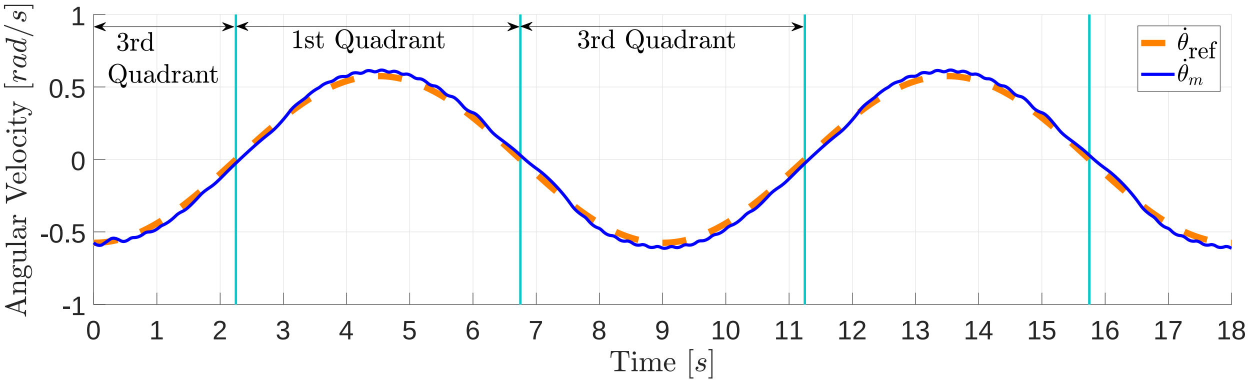


Figure 7.26: Sinusoidal angular velocity

If the velocity tracking presented in figure 7.26 was to be used to counteract wave motion, the payload movement would be as presented in figure 7.27. Here, the system manages to compensate the payload within ±10cm. However, positioning accuracy would be improved if the implemented angular velocity controller was extended to feature position control.

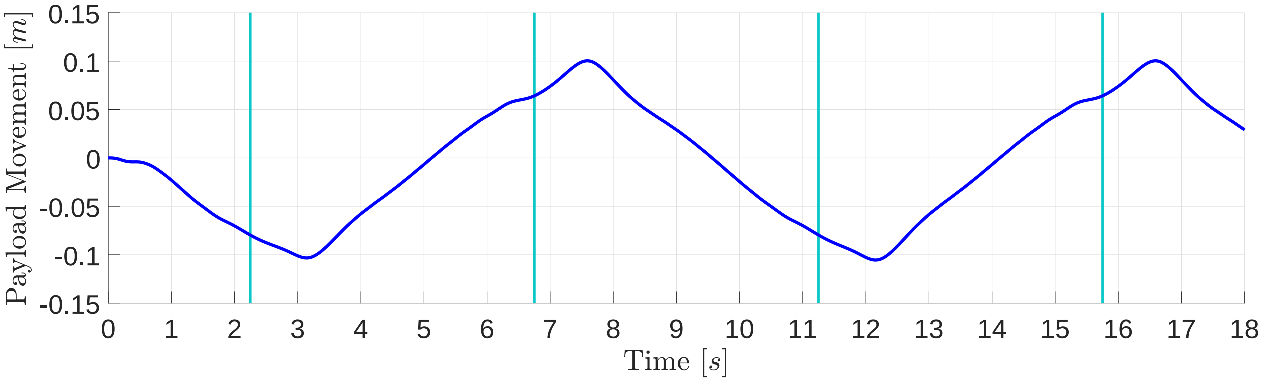


Figure 7.27: Payload movement during active heave compensation

Figure 7.28 presents the displacement parameter α during the sinusoidal velocity tracking. As can be seen, the displacement is changed in order to produce the motor torque T_m in figure 7.29.

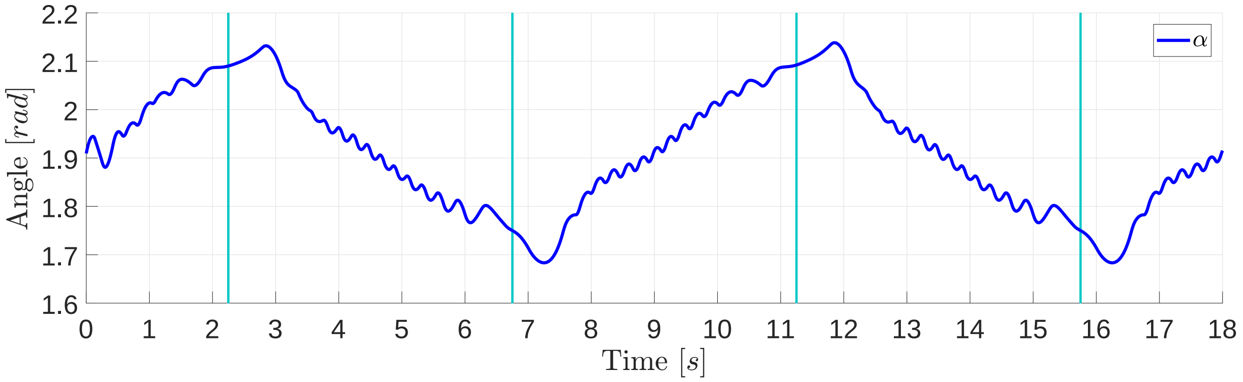


Figure 7.28: Displacement parameter α during heave compensation

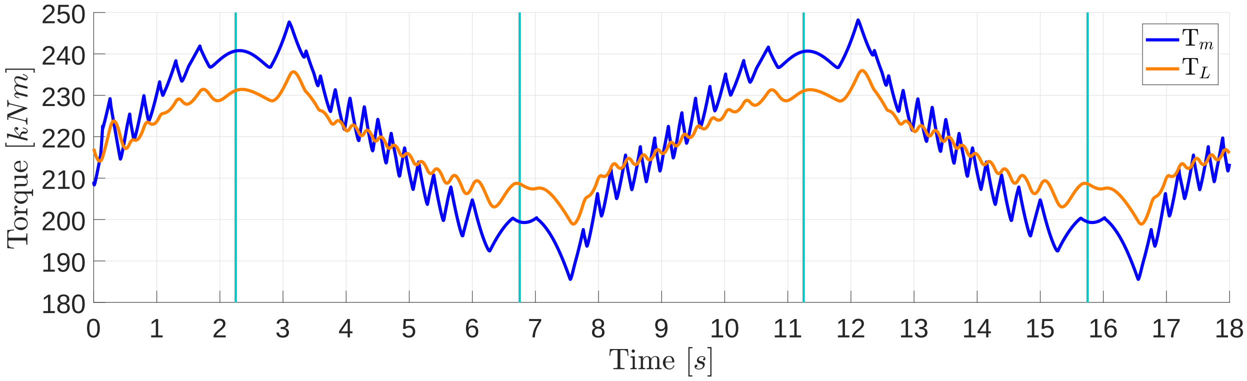


Figure 7.29: Motor- and load torque during heave compensation

Resultant oil flows in to and out of the motor performing heave compensation are presented in figure 7.30. The instant change in oil flow at $t = 0s$ is due to the initial velocity of the motor. During the entire heave compensation sequence, no flow peaks are observed.

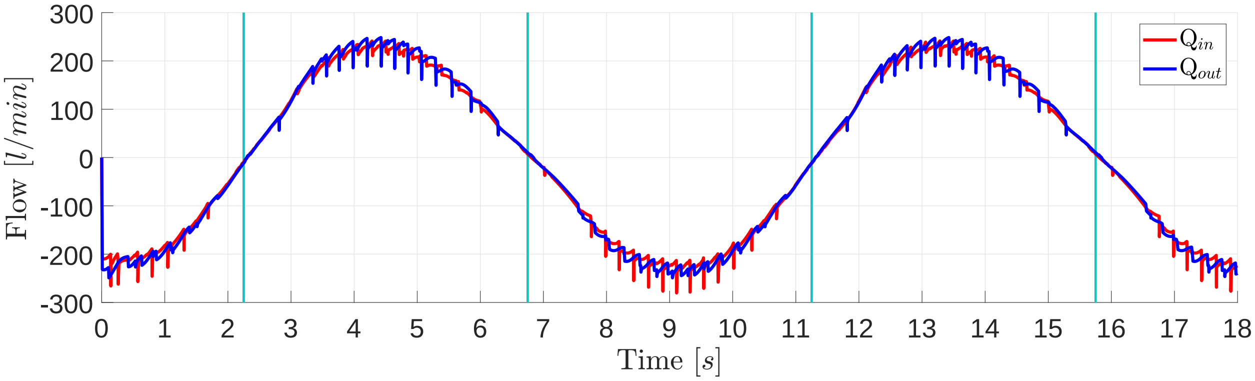


Figure 7.30: Motor flows during heave compensation

8 | Discussion

The discussion chapter is structured into smaller sections that discusses assumptions made while modelling the system, the validation process of the suggested control system and different aspects of the presented results.

General Valve Timing and Cylinder Behaviour

The valve timing strategy in CCW-motor mode and CW-pump mode (1st and 3rd quadrant) works as intended. Operation in the second quadrant (CW-motor) has been tested to a lesser extent, but the general behaviour of the valves and the corresponding pressure levels in the cylinders show expected results. Tests of operation in the fourth quadrant (CCW-pump) has not been carried out.

The general strategy applied for compression and decompression of the fluid also works as intended. When the sequence is initiated, the pressure levels in the cylinder chambers are correctly pressurised and the valves open against a low pressure difference. At the onset of opening the HPV, the chamber pressure is slightly higher than that of the high pressure line ($350-360bar$). Similar differences are observed at the onset of opening the LPV when the chamber has been depressurised. The resulting small flow peaks are observed over the valves. This is largely due to the assumption mentioned in section 2.6 that the pressure gradient is zero as the valves open/close. The valve timing is also slightly affected by the motor shaft velocity. However, if the motor operates at higher velocities than the values that were used to dimension the valves, the pressurisation/depressurisation sequences will be affected, as the piston will travel further during the closing/opening of the valves.

Choice of Displacement Strategy

Even though the efficiency of the system has not been analysed, it is expected that the partial stroke displacement strategy at some extent decreases the efficiency compared to full stroke displacement, especially as the cylinders are allowed to reactivate during motor modes, and deactivate before TDC during pump modes.

If the controller that determines the displacement parameter gives a relatively stable output, the control system will not reactivate/deactivate many cylinders in motor and pump mode respectively. As these actions require that the valves open against high pressure differences, the efficiency of the system is dependent on how well-regulated the displacement parameter is.

In terms of torque response, the choice of displacement strategy has proved to be efficient. The motor torque changes almost simultaneously with the displacement parameter, slightly delayed by the time used for pressure build-up.

As a consequence of the chosen displacement strategy, the implementation of a feedback PID-controller to determine the displacement parameter is fairly straight forward, in contrast to a full stroke displacement strategy that demands more complex control system architecture. More intricate control theory can of course be applied to improve the reference tracking, but this has not been a primary focus in this thesis.

Operation Mode Controller and Mode Transitions

The use of the sign of the displacement parameter α paired with the velocity reference to determine the load scenario gives satisfactory results. This is partially due to the feed-forward element, which counters for the most of the steady-state load. The controller error compensates therefore mainly for the dynamical behaviour of the system. Without a feed-forward, the controller would possibly fluctuate more often around zero, thus frequently changing between operation modes.

During the validation of the control system, the DDM performs well when switching between the two most important quadrants (first and third). This is because the load is stable, yielding low changes in the displacement parameter when the velocity reference is reversed.

When the load is manipulated to provoke the control system to switch from the first to the second quadrant in section 6.3.1, the control parameter is forced to have a value of zero for a small period before the mode switch. This is because the actual value is negative (because of the overshooting velocity error), which in turn together with a positive velocity reference means that the control system would have chosen to operate in the fourth quadrant (CCW-pump). This scenario is similar to the one described as a motivation for developing a valve timing strategy for the fourth quadrant in section 3.3.1. However, this is a short interval, and the DDM recovers quickly as soon as it has switched to the second quadrant. Implementation of a fourth quadrant operation mode has therefore not been deemed necessary.

The test results from the mode switching from the third to the second quadrant shown in section 6.3.2 highlight some challenges. The control system behaves as expected in the respective modes. But the manipulated load in the test leads to changes to the displacement parameter that forces cylinders to deactivate before TDC in CW-pump mode, and reactivate in CW-motor mode. These situations are unavoidable as long as the valves are allowed to open against high pressure differences.

The test results from the mode switching from the second to the third quadrant shown in section 6.3.2 shows that the cylinders are not particularly affected by the change in operation mode.

Lock Mode

Prior to implementation of the lock mode, the digital motor struggled to follow a velocity reference of $0 rad/s$. Also, the oscillations translated to a degraded velocity reference tracking in the following other regions. With lock mode activated, the system shows promising results. The payload did not drop down as lock mode was exited, which is considered important. Also, the velocity reference tracking was greatly improved with lock mode enabled. Due to the lock mode, the payload is practically at rest when the velocity reference starts to increase from zero velocity. However, as the motor exits lock mode, some flow peaks occur. This is due to the small oscillations stabilising at a point where the pressure levels in the chambers are either slightly higher or slightly lower then they were when entering the lock mode. As such, some cylinders opens against a higher pressure difference than predicted by the control system.

Applied Load Case

The system shows promising results when operating in the different operation modes, with out significant pressure- or flow peaks during operation. Further, the winch system shows satisfactory reference tracking with the selected control strategy for the tested load cases. The assumption made in section 6.3 that the control system will not switch directly from the second quadrant to the first without transitioning through the third quadrant is partially proved when the light load is lowered in section 7.2.2. The control system only occasionally chooses to operate in the second quadrant if it struggles to follow a downwards velocity reference.

9 | Conclusion

The potential in increased efficiency of hydraulic systems actuated by DDMs is normally the main motivation for research in this field. In this thesis, focus has been aimed at highlighting advantages and challenges when implementing this technology to a typical industrial winch application. As such, efficiency demands has not been a central part of the presented work. The suggested control strategy can form the basis for an efficency analysis if leakages over the valves and cylinders and friction in the cylinders are introduced. In that case, the choice of letting the cylinders reactivate/deactivate must be weighted up against the reference tracking accuracy of the system, especially in cases where accurate load handling is a crucial operational factor.

When applying special conditions to ensure a smooth transition between the different modes, focus has been aimed on ensuring safe operation of the motor. Thus, measures to prevent high pressure build-up or cavitation have been implemented. However, all possible scenarios when modes are switched are not investigated and further work should be done to ensure that no dangerous operational situation can occur.

If desired, cylinder behaviour during the mode switches can further be optimized with respect to either efficiency or system response. The trade-off between the two being at what extent the valves should be able to open against high pressure differences.

In particular, a strategy for the switch from the third to the second quadrant (and correspondingly from the fourth to the first, if operation in the fourth quadrant is considered) is important to determine. Strategies could include letting the cylinders finish their cycle through the mode switch if efficiency is important, letting them reactivate/deactivate as shown if accuracy is more important, or switching directly from the first to the second quadrant without going through the third if a light load is connected to the winch.

Operation in the fourth quadrant has not been deemed necessary, as the load cases tested with the winch system have not produced any scenario where the control system chose to operate as a pump in the CCW-direction. However, in a small interval during stress testing of the control system with a manipulated load, the conditions for operation in the fourth quadrant were fulfilled. Apart from increased accuracy in a small time interval, operation in the fourth quadrant could be beneficial for the total system efficiency as energy would have been fed back to the system.

Simulations of the winch system shows promising results and gives the desired performance in terms of system behaviour and handling of the load. The lock mode introduced for steady handling of the payload at a velocity reference of $0rad/s$ also shows promising results. This mode was developed as the reference tracking challenge around $0rad/s$ presented itself during testing, and should therefore be further addressed and tested to discover possible advantages and challenges.

Bibliography

[1] K. J. Merrill, M. A. Holland, and J. H. L. Jr., "Efficiency analysis of a digital pump/motor as compared to a valve plate design," *7th International Fluid Power Conference*, 2010.

[2] C. Nørgård, "Design, optimization and testing of valves for digital displacement machines," PhD thesis, Aalborg University, 2017.

[3] T. O. Andersen and M. R. Hansen, "Hydraulic components and systems," Grimstad, Aug. 2012.

[4] B. R. AG, "Rexroth 4ee rexroth for energy efficiency," Accessed 23/1-18, 2011. [Online]. Available: https://www.boschrexroth.com/en/xc/trends-and-topics/energy-efficiency/index

[5] M. R. Hansen, "Course notes: Mas219 hydraulics, university of agder, norway," 2015.

[6] S. W. Rampen, Artemis IP Ltd, "Gearless transmissions for large wind turbines," *The history and Future of Hydraulic Drives*, 2006.

[7] D. B. Roemer, "Design and optimization of fast switching valves for large scale digital hydraulic motors," PhD thesis, Aalborg University, 2014.

[8] M. R. Hansen, "Fluid mechanics," 04 2010.

[9] D. B. Rømer, P. Johansen, H. C. Pedersen, and T. O. Andersen, "Analysis of valve requirements for high-efficiency digital displacement fluid power motors," *Conference: Proc. 8th International Conference on Fluid Power Transmission and Control (ICFP2013)*, 2013.

[10] BoschRexroth, "Secondary control," *Drive and Control*, vol. 01, 2014. [Online]. Available: https://dc-corp.resource.bosch.com/media/general_use/products/systems/secondary_controls/dokumentation_and_downloads/Secondary_Control_Drive_and_Control_12014_.pdf

[11] S. W. Rampen, Artemis IP Ltd, "The development of digital displacement technology," *Fluid Power and Motion Control*, 2010.

[12] P. Calzoni, "Radial piston motor, type mrd, mrde, mrv, mrve," datasheet.

A | MATLAB Scripts and Functions

A.1 Script: Winch Model With Ideal Motor

```
close all
clear


%%  -- Inputs --

%system specifications
zdot_pl_max = 1.5;
z_pl_ini = 3660;
w_frac = 0.5; %amount of wire after spring

acc_wave = 1*(2*pi/9)^2;

%%  -- Stuff --

%simulation parameters
dt=1e-5;                        %Step time              [s]
t = 0;                          %Start time             [s]
T = 15;                         %End time               [s]
counter = 1;                    %Plot counter

%motor and hydraulic parameters
pH = 350e5;                     %High pressure side     [Pa]
pL = 5e5;                       %Low pressure side      [Pa]

%%  -- Initialisation --

%drum
teta_shaft = 0;                 %inital shaft position  [rad]
tetadot = 0*pi/30;              %inital shaft velocity  [rad/s]

%payload
z_pl = z_pl_ini;                %inital payload position [m]
zdot_pl = 0;                    %inital payload velocity [m/s]

%%  -- Load --

rho_sw  = 1030;                 %Sea water density      [kg/m^3]
rho_st = 7850;                  %Steel density          [kg/m^3]
g = 9.81;                       %gravitational acc      [m/s^2]

dw = 34e-3;                     %diameter wire          [m]
dd = 1632e-3;                   %diameter drum          [m]
```

```
wd = 1260e-3;                   %width drum             [m]
rho_w = 5.66;                   %wire density           [kg/m]
A_w = pi*dw^2/4;                %Wire area              [m^2]
L = z_pl_ini;                   %Length of wire out     [m]
m_wout = L*rho_w;               %Spooled out wire mass  [kg]
m_pl = 20e3;                    %20 tonne payload       [kg]
V_pl = 2.55;                    %Payload volume         [m^3]

E = 210e9;                      %E-modulus wire         [Pa]
k_w = E*A_w/L;                  %Wire stiffness         [N/m]
b_w = 0.1*k_w;                  %Wire damping           [N*s/m]

Fb_pl = V_pl*rho_sw*g;          %Payload buoyancy       [N]
Fb_w = L*A_w*rho_sw*g;          %Wire buoyancy          [N]
delta0 = ( (m_pl + m_wout*w_frac)*g ...
    - Fb_pl - Fb_w*w_frac )/k_w; %static wire deflection [m]

%initial conditions
n_tmax = floor(wd/dw);          %number of turns per layer [-]
n_l = 1;                        %current wire layer     [-]
L_wd = 0;                       %length of wire on drum [m]

while true
    r_eff = dd/2 + dw + (n_l - 1)*0.8*dw - dw/2; %calculate effective drum radius
    L_wt = 2*pi*r_eff;         %Length of wire per turn for current layer
    if mod(n_l,2)              %number of turns for odd layers
        n_t = n_tmax;
    else                       %number of turns for even layers
        n_t = n_tmax - 1;
    end
    for current_n_t = 1:n_t    %increase wire length per turn
        L_wd = L_wd + L_wt;
        if L_wd >= 3660 - L
            break
        end
    end
    if L_wd >= 3660 - L
        break
    end
    n_l = n_l + 1;             %increase number of layers
end

%to simulation
tetadotdot_dem = acc_wave/r_eff;

%determination of vicious friction coefficient
n_c = 42;
Vd = 1400*1e-6;
Tmax = 1*n_c*Vd*(pH-pL)/2/pi;
tetadot_max = zdot_pl_max/r_eff;
bb = 0.01*Tmax/tetadot_max;     % 1% of Tmax @ tetadot_max

%%  -- Memory allocation --

interval = 1000;
plot_counter = interval;
```

```
steps = floor(T/dt/interval);

t_plot = zeros(steps,1);
teta_plot = zeros(1,steps);
tetadot_plot = zeros(steps,1);
tetadotdot_plot = zeros(1,steps);
z_pl_plot = zeros(1,steps);
zdot_pl_plot = zeros(1,steps);
delta_plot = zeros(1,steps);
deltadot_plot = zeros(1,steps);
Fw_plot = zeros(1,steps);
TL_plot = zeros(1,steps);

%%  -- Simulation --

tic

while t<T

        %______________MECHANICAL MODEL___________________

    delta = r_eff*teta_shaft - (z_pl - z_pl_ini) + delta0;
    deltadot = r_eff*tetadot - zdot_pl;

    if delta <= 0
        Fw = 0;     %Wire strekk only valid for strekk
    else
        Fw = delta*k_w + deltadot*b_w;
    end

    TL = (Fw + m_wout*(1-w_frac)*g-Fb_w*(1-w_frac))*r_eff + bb*tetadot;

    zdotdot_pl = (Fw + Fb_pl + Fb_w*w_frac ...
        - (m_pl + m_wout*w_frac)*g)/(m_pl + m_wout*w_frac);

    %KINEMATIC DRIVER:
   if t < zdot_pl_max/r_eff/tetadotdot_dem
       tetadotdot = tetadotdot_dem;
   else
       tetadotdot = 0;
   end

    %plotting
    if plot_counter == interval
        plot_counter = 0;

        t_plot(counter) = t;
        teta_plot(counter) = teta_shaft;
        tetadot_plot(counter) = tetadot*30/pi;
        tetadotdot_plot(counter) = tetadotdot;
        z_pl_plot(counter) = z_pl;
        zdot_pl_plot(counter) = zdot_pl;
        delta_plot(counter) = delta;
        deltadot_plot(:,counter) = deltadot;
        Fw_plot(counter) = Fw;
        TL_plot(counter) = TL;
        counter = counter + 1;
    end

        %_______________ITERATIONS & MISC____________________

    %Motion
    teta_shaft = teta_shaft + tetadot*dt;
    tetadot = tetadot + tetadotdot*dt;
    z_pl = z_pl + zdot_pl*dt;
    zdot_pl = zdot_pl + zdotdot_pl*dt;

    %Simulation
    t = t + dt;
    plot_counter = plot_counter + 1;

end

toc
%%  -- Results --
T_dem = max(TL_plot);
print_demand = fprintf('\n Demanded torque : %.0f Nm\n', T_dem);

%%  SHAFT
% figure('Name','Shaft ','NumberTitle','Off')
%
% tetadot_ref_plot=zdot_pl_max/r_eff*ones(1,length(t_plot));
%
%     subplot(2,1,1) %Velocity
% title('Shaft Angular Velocity')
% plot(t_plot,tetadot_plot*pi/30,'b',t_plot,tetadot_ref_plot,'r--')
% grid
% hl = legend('$\dot{\theta}$','$\dot{\theta}_{\textnormal{ref}}$');
% set(hl, 'Interpreter', 'latex');
% ylabel('Angular Velocity [rad/s]')
%
%     subplot(2,1,2) %Angle
% title('Shaft Angle')
% plot(t_plot,teta_plot,'b')
% grid
% legend('\theta')
% ylabel('Angle [rad]')
% xlabel('Time [s]')

%% Payload
% figure('Name','Payload','NumberTitle','Off')
%
% zdot_pl_ref_plot = tetadot_ref_plot*r_eff;
%
%     subplot(2,1,1) %Velocity
% title('Payload Velocity')
% plot(t_plot,zdot_pl_plot,'b',t_plot,zdot_pl_ref_plot,'r--')
% grid
% hl = legend('$\dot{z}_{\textnormal{pl}}$','$\dot{z}_{\textnormal{pl,ref}}$');
% set(hl, 'Interpreter', 'latex');
%
```

```
%     subplot(2,1,2) %Position
% title('Payload Position')
% plot(t_plot,z_pl_plot,'b')
% grid
% legend('z_{pl}')

%%  TORQUE
figure('Name','Torque','NumberTitle','Off')

Tmax_plot = Tmax*ones(1,length(t_plot));

    subplot(2,1,1) %Torque
plot(t_plot,TL_plot/1e3,':','LineWidth',1.2,'Color',[0, 0.7, 0])
grid
ylabel('Torque [kNm]')
hl = legend('$\textnormal{T}_{L}$');
set(hl, 'Interpreter', 'latex');
title('Torque')

    subplot(2,1,2) %Acceleration
plot(t_plot,tetadotdot_plot,'b')
grid
ylabel('Angular Acceleration [s^{-2}]')
xlabel('Time [s]')
hl = legend('$\ddot{\theta}$');
set(hl, 'Interpreter', 'latex');
title('Drum Acceleration')

%%  Fd
% figure('Name','Wire force','NumberTitle','Off')
%
% plot(t_plot,Fw_plot,'b')
% grid
% title('Wire Force')
% legend( 'F_{w}' )

%% Deltas
% figure('Name','Deltas','NumberTitle','Off')
%
% delta0_plot = delta0*ones(1,length(t_plot));
%
%     subplot(2,1,1) %delta
% plot(t_plot,delta_plot,'b')
% hold on
% plot(t_plot,delta0_plot,'r')
% grid
% legend('\delta','\delta_0')
% title('\delta')
%
%     subplot(2,1,2) %deltadot
% plot(t_plot,deltadot_plot,'b')
% grid
% hl = legend('$\dot{\delta}$');
% set(hl, 'Interpreter', 'latex');
% hl = title('$\dot{\delta}$');
% set(hl, 'Interpreter', 'latex');
```

A.2 Script: Motor Selection

```
clear
close all

%% FROM SYSTEM SPECS

pH = 350e5;                        %high pressure side [Pa]
pL = 5e5;                          %low pressure side [Pa]
zdot_pl_ref = 1.5;                 %payload reference velocity [m/s]
z_pl_ini = 3660;                   %initial payload position [m]
deltap = 1.21e5;                   %pressure difference @ demanded flow [Pa]

A_wave = 1;                        %wave amplitude [m]
T_wave = 9;                        %wave period [s]

if z_pl_ini == 200
    TL = 235711;                   %from simulation with kinetic driver @200m
elseif z_pl_ini == 3660
    TL = 308173;                   %from simulation with kinetic driver @3660m
end

%% Effective radius and drum inertia

Jd = 1111;                         %Inertia drum [kg*m^2]
Jm = 9.68;                         %Inertia motor [kg*m^2]

dw = 34e-3;                        %diameter wire [m]
dd = 1632e-3;                      %diameter drum [m]
wd = 1260e-3;                      %width drum [m]
rho_w = 5.66;                      %wire density [kg/m]

L_wout = z_pl_ini;                 %Length of spooled out wire [m]
m_wout = L_wout*rho_w;             %Mass of spooled out wire [kg]

%initial conditions
n_tmax = floor(wd/dw);             %number of turns per layer [-]
n_l = 1;                           %current wire layer [-]
L_wd = 0;                          %length of wire on drum [m]

while true
    r_eff = dd/2 + dw + (n_l - 1)*0.8*dw - dw/2; %calculate effective drum radius
    L_wt = 2*pi*r_eff;            %Length of wire per turn for current layer
    if mod(n_l,2)                 %number of turns for odd layers
        n_t = n_tmax;
    else                          %number of turns for even layers
        n_t = n_tmax - 1;
    end
    for current_n_t = 1:n_t       %increase wire length per turn
        L_wd = L_wd + L_wt;
        if L_wd >= 3660 - L_wout
            break
        end
    end
    if L_wd >= 3660 - L_wout      %end criteria
        break
    end
    n_l = n_l + 1;                %increase number of layers
end

L_wd = 3660 - L_wout;             %acocunt for r_eff script which calculates per turn
m_wd = rho_w*L_wd;                %mass of wire on drum
Jeff = Jm + Jd + m_wd/2*( dd^2/4 + r_eff^2 ) + m_wout/2*r_eff^2; %effective inertia
acc_dem = (2*pi/T_wave)^2*A_wave; %wave acceleration amplitude
tetadot_ref = zdot_pl_ref/r_eff;  %reference anguluar velocity

%% MOTOR AND VALVE SELECTION

Vd_vector = (100:50:10000)*1e-6;  %range of available displacement volumes
n_c = 42;                         %fixed number of cylinders
T_dem = TL + acc_dem/r_eff*Jeff;  %torque demand

Vd_dem = 2*pi*T_dem/n_c/(pH-pL);
index = find(Vd_vector >= Vd_dem,1);
Vd = Vd_vector(index);

Q_avg = tetadot_ref*30/pi*Vd*1e3; %average flow
Q_dem = Q_avg*pi;                 %maximum flow
kf = sqrt(deltap)/(Q_dem/60000);  %required kf

%to simulation model:
print_demand0 = fprintf('\n___________________RESULTS:___________________\n');
print_demand1 = fprintf(...
    '\n Required displacement: %.0f cc \n', Vd*1e6);
print_demand2 = fprintf(...
    '\n Required flow-pressure coefficient : %.0f s*Pa^0.5/m^3 \n \n', kf);
```

A.3 Script: Main script

```
close all
clear

%%  -- Inputs --

%--Plots: 0=off, 1=on
BIGR_INFO_PLOTS = 1;
interval    = 50;                %Plot every n-th value
s1          = 40;                %Plot cylinder #s
s2          = 31;
s3          = 41;

%selected motor parameters:
Vd          = 1400*1e-6;        %Cylinder displacement [m^3]
n_c         = 42;               %Number of cylinders
kf          = 275961;           %Valve flow-pressure coefficient [sqrt(Pa)*s/m^3]
r_eff       = 1.2138;           %Effective radius [m]
ts          = 8e-3;             %Valve switching time [s]
charge_time = 30e-3;

%system specifications
zdot_pl_max = 1.5;              %Payload maximum velocity [m/s]
z_pl_ini    = 200;
zdot_pl_ini = 0*zdot_pl_max*0.8;
w_frac      = 0.5;              %amount of wire after spring
tetadot_max = zdot_pl_max/r_eff;%Highest angular velocity
Tw          = 9;%21;            %Wave period [s]


loaded      = 1;                %winch on/off
IMPROVED    = 1;                % safety switch on/off
IMPROVED2   = 1;
IMPROVED3   = 1;
IMPROVED4   = 1;
LOCKED      = 1;

%%  -- Stuff --

%simulation parameters
dt=5e-5;                        %Step time [s]
t = 0;                          %Start time [s]
counter = 1;                    %Plot counter

%motor and hydraulic parameters
V0 = Vd;                        %Cylinder dead volume [m^3]
pH = 350e5;                     %High pressure side [Pa]
pL = 5e5;                       %Low pressure side [Pa]
Tmax = n_c*Vd*(pH-pL)/2/pi;     %Maximum motor torque [Nm]
betaL = 16000e5;                %oil bulk modulus [Pa]
beta = betaL;                   %inital bulk modulus [Pa]
```

```
epsilon_g0 = 0.002;             %0.2 % gas in oil @ atmospheric pressure
kappa = 1.4;                    %Adiabatic coefficient [-]
p_atm = 101350;                 %Atmospheric pressure [Pa]
Jm = 9.68;                      %Motor inertia [kgm^2]
Jd = 1111;
load data_and_func/valve_dyn_poly
poly_valve=fit_pos.coeff;       %polynomial curve fitting for valve dynamics

%%  -- Controller --


    %Velocity parameters
Kp_vel =  1.34;
Ti_vel = 0.86;
Td_vel = 0.015;

if loaded
    Kp_vel = 1.7;
    Ti_vel = 0.85;
    Td_vel = 0;
end

    %Determine alpha_max and incrementation tolerance
teta2 = pi;
V2 = V0 + Vd/2*(1 - cos(teta2));
RHS_int= int_var_beta(pH,pL,betaL,epsilon_g0,p_atm,kappa);
V1 = exp(log(V2) - RHS_int);
teta1 = acos(2/Vd*(V0-V1)+1);
alpha_max = teta1 - 0;

    %alpha feed forward:
load('data_and_func/alpha_ff_poly')     %alpha_ff_func
alpha_poly = alpha_ff_poly.coeff;       %alpha_ff_poly%poly_alpha_ff

%% -- Reference signal --

%parameters delimiting the periods of the reference:
t0 = 10;
t1 = t0 + Tw/2;
t2 = t1 + 10;
t3 = t2 + Tw/4;
t4 = t3 + Tw*2;
t5 = t4 + Tw/2;
T = t4;

    %velocity values for the different periods:
start_vel0=-10*pi/30*0;                 %Starting velocity      [m/s]
%part 0
end_vel0=start_vel0;                    %End Velocity           [m/s]
%part 1
end_vel1=1*end_vel0;                    %part 1
%part 2
end_vel2=end_vel1;                      %part 2
%part 3
end_vel3=0*-1*end_vel2;                 %part 3
%part 4
```

```
end_vel4=end_vel3;                      %part 4
%part 5
end_vel5=end_vel4;                      %part 5



if T >= t0 % part 0
    start_time = 0;                                   %Time at start of path   [s]
    total_time = t0;                                  %Total path time         [s]
    q = [start_vel0; 0; end_vel0; 0];                 %motion vector
    time = [start_time; (start_time+total_time)];     %time interval
    [vel, tt] = vel_profile(q,time,dt,total_time);    %see vel_profile.m for details
    VelGen = [tt', vel'];
end

if T >= t1 % part 1
    start_time = t0;
    total_time = t1-t0;
    start_vel1=end_vel0;% + 0.01;
    q = [start_vel1; 0; end_vel1; 0];
    time = [start_time; (start_time+total_time)];
    [vel, tt] = vel_profile(q,time,dt,total_time);
    VelGen = [VelGen; tt', vel'];
end
if T >= t2 % part 2:
    start_time = t1;
    total_time = t2-t1;
    start_vel2=end_vel1;
    q = [start_vel2; 0; end_vel2; 0];
    time = [start_time; (start_time+total_time)];
    [vel, tt] = vel_profile(q,time,dt,total_time);
    VelGen = [VelGen; tt', vel'];
end
if T >= t3 % part 3:
    start_time = t2;
    total_time = t3-t2;
    start_vel3=end_vel2;
    q=[start_vel3; 0; end_vel3; 0];
    time = [start_time; (start_time+total_time)];
    [vel, tt] = vel_profile(q,time,dt,total_time);
    VelGen = [VelGen; tt', vel'];
end
if T >= t4 % part 4:
    start_time = t3;
    total_time = t4-t3;
    start_vel4=end_vel3;
    q = [start_vel4; 0; end_vel4; 0];
    time = [start_time; (start_time+total_time)];
    [vel, tt] = vel_profile(q,time,dt,total_time);
    VelGen = [VelGen; tt', vel'];
end
if T >= t5 % part 5:
    start_time = t4;
    total_time = t5-t4;
    start_vel5=end_vel4;
    q = [start_vel5; 0; end_vel5; 0];
    time = [start_time; (start_time+total_time)];
    [vel, tt] = vel_profile(q,time,dt,total_time);
    VelGen = [VelGen; tt', vel'];
end

VelGen = [VelGen; tt(end)*ones(1,20)' vel(end)*ones(1,20)'];
index = 1;

%% -- Visualize path --

% figure('Name','Velocity profile','NumberTitle','Off');
% plot(VelGen(:,1),VelGen(:,2),'r');
% grid;
% legend('vel');
% xlabel('time [s]');
% ylabel('angular velocity [s^{-1}]');

%% -- Load --

rho_sw  = 1030;                 %Sea water density      [kg/m^3]
rho_st = 7850;                  %Steel density          [kg/m^3]
g = 9.81;                       %gravitational acc      [m/s^2]

dw = 34e-3;                     %diameter wire          [m]
dd = 1632e-3;                   %diameter drum          [m]
wd = 1260e-3;                   %width drum             [m]
rho_w = 5.66;                   %wire density           [kg/m]
A_w = pi*dw^2/4;                %Wire area              [m^2]
L = z_pl_ini;                   %Length of wire out     [m]
m_wout = L*rho_w;               %Spooled out wire mass  [kg]
m_pl = 20e3;                    %20 tonne payload       [kg]
V_pl = 2.55;                    %Payload volume         [m^3]

E = 210e9;                      %E-modulus wire         [Pa]
k_w = E*A_w/L;                  %Wire stiffness         [N/m]
b_w = 0.1*k_w;                  %Wire damping           [N*s/m]

Fb_pl = V_pl*rho_sw*g;          %Payload buoyancy       [N]
Fb_w = L*A_w*rho_sw*g;          %Wire buoyancy          [N]
delta0 = ( (m_pl + m_wout*w_frac)*g ...
    - Fb_pl - Fb_w*w_frac )/k_w;%static wire deflection [m]

    %initial conditions
n_tmax = floor(wd/dw);          %number of turns per layer [-]
n_l = 1;                        %current wire layer     [-]
L_wd = 0;                       %length of wire on drum [m]

while true
    r_eff = dd/2 + dw + (n_l - 1)*0.8*dw - dw/2; %calculate effective drum radius
    L_wt = 2*pi*r_eff;         %Length of wire per turn for current layer
    if mod(n_l,2)              %number of turns for odd layers
        n_t = n_tmax;
    else                       %number of turns for even layers
        n_t = n_tmax - 1;
    end
    for current_n_t = 1:n_t    %increase wire length per turn
```

```
        L_wd = L_wd + L_wt;
        if L_wd >= 3660 - L
            break
        end
    end
    if L_wd >= 3660 - L
        break
    end
    n_l = n_l + 1;             %increase number of layers
end

m_wd = (3660 - z_pl_ini)*rho_w;

Jeff = Jm + Jd + m_wd/2*( dd^2/4 + r_eff^2 ) + m_wout/2*r_eff^2; %effective inertia
bb = 1793;

if loaded
    TL = (delta0*k_w + m_wout*(1-w_frac)*g-Fb_w*(1-w_frac))*r_eff;
else
    TL = -0.8*Tmax;
end

%% -- Initialisation --

    %motor
deltateta = 2*pi/n_c;           %cylinder phase angle [rad]
teta_shaft = 0;                 %inital shaft position [rad]
tetadot = start_vel0;           %inital shaft velocity [rad/s]
alpha_ff  = sign(TL)*find_alpha_ff(abs(TL),alpha_poly);%determine initial alpha [rad]
alpha = alpha_ff;               %inital closing angle [rad]
    %cylinder TDC and pressure:
teta = zeros(1,n_c);            %seperate cylinder angle [rad]
pc = pL*ones(1,n_c);            %cylinder chamber pressure [Pa]
xH = zeros(1,n_c);              %valve plunger position [-]
xL = ones(1,n_c);               %valve plunger position [-]
t_HPV = zeros(1,n_c);           %time variable for valve dynamics [s]
t_LPV = ones(1,n_c)*ts;         %time variable for valve dynamics [s]
if sign(alpha) == 1
    for ii = 1:n_c
        teta(ii) = 2*pi - deltateta*(ii-1) + teta_shaft;%start seperate cylinder angle @ T
        if teta(ii) <= alpha || teta(ii) == 2*pi
            pc(ii) = pH;        %pressurize cylinders
            xH(ii) = 1;         %initialize valves accordingly
            xL(ii) = 0;
            t_HPV(ii) = ts;
            t_LPV(ii) = 0;
        end
    end
elseif sign(alpha) == -1
    alpha = 2*pi - abs(alpha);
    for ii = 1:n_c
        teta(ii) = 2*pi - deltateta*(ii-1);%start seperate cylinder angle @ TDC
        if teta(ii) >= alpha || teta(ii) == 2*pi
            pc(ii) = pH;        %pressurize cylinders
            xH(ii) = 1;         %initialize valves accordingly
            xL(ii) = 0;
            t_HPV(ii) = ts;
            t_LPV(ii) = 0;
        end
    end
end

LPVb = xL;                      %control system booleans
HPVb = xH;                      %          ''

    %special case initialisations
if sign(end_vel0) == 1 || sign(end_vel0) == 0 && sign(end_vel1) == 1
    pc(2) = 274.12e5;
    xL(2) = 0;
    LPVb(2) = 0;
    t_HPV(2) = 0;
    t_LPV(2) = 0;
end
if sign (alpha_ff) == -1
    pc(41) = (1-0.9441)*pH;
    pc(42) = (1-0.2166)*pH;
    xL(41:42) = 0;
    LPVb(41:42) = 0;
    t_HPV(41:42) = 0;
    t_LPV(41:42) = 0;
end

LPV_closing = zeros(1,n_c);     %control system booleans
LPV_opening = zeros(1,n_c);
HPV_closing = zeros(1,n_c);
HPV_opening = zeros(1,n_c);

tetaHPVo_p = zeros(1,n_c);      %initialisation of angles
tetaLPVo_m = ones(1,n_c)*pi;
tetaHPVc_p = 0;

    %special cases for switching
tetaLPVo_switch = zeros(1,n_c);
tetaHPVo_switch = zeros(1,n_c);
change_31 = zeros(1,n_c);
change_13 = zeros(1,n_c);
change_12 = zeros(1,n_c);
change_23 = zeros(1,n_c);


    %payload
z_pl = z_pl_ini;                %inital payload position [m]
zdot_pl = tetadot*r_eff;        %inital payload velocity [m/s]

%% -- Memory allocation --


plot_counter = interval;        %interval counter
steps = floor(T/dt/interval);   %plotting dimension

t_plot = zeros(steps,1);
tetadot_plot = zeros(steps,1);
```

```
Tm_plot = zeros(1,steps);
TL_plot = zeros(1,steps);
pc_plot = zeros(n_c,steps);
xH_plot = zeros(n_c,steps);
xL_plot = zeros(n_c,steps);
QH_plot = zeros(n_c,steps);
QL_plot = zeros(n_c,steps);
e_teta_plot = zeros(1,steps);
teta_plot = zeros(1,steps);
Vc_plot = zeros(n_c,steps);
Vcd_plot = zeros(n_c,steps);
teta_cyl = zeros(n_c,steps);
alpha_plot = zeros(1,steps);
xp_plot = zeros(n_c,steps);
tetadotdot_plot = zeros(1,steps);
Qin_plot = zeros(1,steps);
Qout_plot = zeros(1,steps);
e_tetadot_plot = zeros(1,steps);
tetadot_ref_plot = zeros(1,steps);
teta_ref_plot = zeros(1,steps);
E_tetadot_plot = zeros(1,steps);
Tc_plot = zeros(n_c,steps);
Vdotsum_plot = zeros(1,steps);
tetaLPVc_plot = zeros(1,steps);
alpha_max_plot = zeros(1,steps);
HPVb_plot = zeros(n_c,steps);
LPVb_plot = zeros(n_c,steps);
HPV_opening_plot = zeros(n_c,steps);
HPV_closing_plot = zeros(n_c,steps);
LPV_opening_plot = zeros(n_c,steps);
LPV_closing_plot = zeros(n_c,steps);
compb_plot = zeros(n_c,steps);
quadrant_plot = zeros(1,steps);
alpha_plot_qu = zeros(1,steps);
tetadot_ctr_plot = zeros(1,steps);
QIN = zeros(1,steps);
E_teta_plot = zeros(1,steps);
tetaHPVo_p_plot = zeros(n_c,steps);
db_plot = zeros(1,steps);
z_pl_plot = zeros(1,steps);
zdot_pl_plot = zeros(1,steps);
delta_plot = zeros(1,steps);
deltadot_plot = zeros(1,steps);
Fw_plot = zeros(1,steps);
quad_line = zeros(1,steps);

%% -- Controller inits --

    %vel controller
e_tetadot = 0;                  %error function angular velocity
E_tetadot = 0;                  %integral error angular velocity
edot_tetadot = 0;
e_tetadot_prev = 0;
alpha_prev = alpha;

%% -- Temp/Debug--
```

```
    %lock-mode parameters:
stop = 0;
Vel_tolerance2 = 0.01;

LoadSwitch_frac = 0.9;         %fraction of velocity ramp time used for load switching
ddt = (t1 - t0)*(1-LoadSwitch_frac)/2;

%% -- Simulation --

tic

while t<T

    %% _____________REFERENCE SIGNAL______________

%     tetadot_ref = VelGen(index,2);
    tetadot_ref = 2*pi/9*sin(2*pi/9*t);

    if t <= t0
        tetadot_ref = 0;
    elseif t > t0 && t <= t1
        tetadot_ref = 0.8*tetadot_max/(t1-t0)*(t - t0);
    elseif t > t1 && t <= t2
        tetadot_ref = 0.8*tetadot_max;
    elseif t > t2 && t <= t3
        tetadot_ref = 0.8*tetadot_max -0.8*tetadot_max/(t3-t2)*(t - t2);
    elseif t > t3
        tetadot_ref = 0;
    end

    %% ________________CONTROLLER_________________

        %error function ang vel
    e_tetadot = tetadot_ref - tetadot;

        %alpha FF
    if alpha_prev ~= alpha
        alpha_ff  = sign(TL)*find_alpha_ff(abs(TL),alpha_poly);
    end

        %alpha expression (PID-controller)
    alpha = Kp_vel*(e_tetadot + 1/Ti_vel*E_tetadot + Td_vel*edot_tetadot) + alpha_ff;

        %Determine alpha_max that ensures decompression/compression:
    teta2 = pi;
    V2 = V0 + Vd/2*(1 - cos(teta2));
    V1 = exp(log(V2) - RHS_int);
    teta1 = acos(2/Vd*(V0-V1)+1);
    alpha_max = teta1 - ts*abs(tetadot);

    %% ____________DETERMINE QUADRANTS____________

    if sign(tetadot_ref) == 0 %special case where tetadot_ref = 0
        if sign(alpha) == 1
            quadrant = 1;
            if alpha >= alpha_max
                alpha = alpha_max;
            end
        elseif sign(alpha) == -1
            quadrant = 2;
            if alpha <= -1*alpha_max
                alpha = -1*alpha_max;
            end
        end
    end

    if sign(alpha) == 1 && sign(tetadot_ref) == 1 || sign(alpha) == -1 && sign(tetadot_ref
        quadrant = 1; %CCW motoring mode

        if alpha >= alpha_max
            alpha = alpha_max;
        elseif alpha < 0
            alpha = 0;
        end

    elseif sign(alpha) == -1 && sign(tetadot_ref) == -1
        quadrant = 2; %CW motoring mode

        if alpha <= -1*alpha_max
            alpha = -1*alpha_max;
        elseif alpha > 0
            alpha = 0;
        end

    elseif sign(alpha) == 1 && sign(tetadot_ref) == -1
        quadrant = 3; %CW pumping mode

        if alpha >= alpha_max
            alpha = alpha_max;
        elseif alpha <= tetaHPVc_p
            alpha = tetaHPVc_p;
        end

        %deactivated, forced to 1st quadrant
%     elseif sign(alpha) == -1 && sign(tetadot_ref) == 1
%         quadrant = 4; %CCW pumping mode
%
%         if alpha > 0
%             alpha = 0;
%         elseif alpha <  -1*alpha_max
%             alpha = -1*alpha_max;
%         end
    end

    if t == 0   %initial value
        quadrant_prev = quadrant;
    end


    %% _____________SWITCHING ANGLES______________
```

```
%///1st quadrant///

    %Compression angles
tetaHPVo_m = 2*pi-ts*abs(tetadot);
V2 = V0 + Vd/2*(1 - cos(tetaHPVo_m));
V1 = exp(log(V2) + RHS_int);
teta1 = 2*pi - acos(2/Vd*(V0-V1)+1);
tetaLPVc_m = teta1 - ts*abs(tetadot);

%/// 2nd quadrant///

if quadrant == 2
    alpha_cw   = 2*pi - abs(alpha);     %mirrored alpha for CW motoring
    tetaLPVc_m = 2*pi - tetaLPVc_m;     %mirrored LPV closing angle
    tetaHPVo_m = 2*pi - tetaHPVo_m;     %mirrored HPV opening angle
end

%/// 3rd quadrant///

    %Compression angles
teta2 = abs(alpha);
V2 = V0 + Vd/2*(1 - cos(teta2));
V1 = exp(log(V2) + RHS_int);
teta1 = acos(2/Vd*(V0-V1)+1);
tetaLPVc_p = teta1 + ts*abs(tetadot);

    %Decompression angles
tetaHPVc_p = ts*abs(tetadot);
V2 = exp(RHS_int + log(V0));
tetaLPVo_p = 2*pi - acos(2/Vd*(V0 - V2) + 1);

%% ______________VALVE TIMING_________________

for ii = 1:n_c

    %% -- Mode switching conditions --


    if IMPROVED3 && quadrant == 2 && quadrant_prev == 1 ...
            && ~HPVb(ii) && ~LPVb(ii) && teta(ii) < pi
        change_12(ii) = 1;
    end

    if IMPROVED4 && quadrant == 3 && quadrant_prev == 2 ...
            && ~HPVb(ii) && ~LPVb(ii) && teta(ii) > pi
        change_23(ii) = 1;
    end

    %% ---CONDITIONS FOR VALVE SWITCHING---
    %booleans determine the optimal valve positions at any given time

        %///1st quadrant - CCW motoring///
    if quadrant == 1

            %HPV
        if teta(ii) >= alpha && teta(ii) <= tetaLPVc_m
```

```
                HPVb(ii) = 0;
            end
            if teta(ii) <= alpha || teta(ii) >= tetaHPVo_m
                HPVb(ii) = 1;
            end

                %LPV
            if ~HPVb(ii) && ~HPV_closing(ii) && xH(ii) == 1
                tetaHPVo_switch(ii) = teta(ii);
                V1 = V0 + Vd/2*(1 - cos(teta(ii) + ts*abs(tetadot)));
                V2 = exp(log(V1) + RHS_int);
                tetaLPVo_m(ii) = acos(2/Vd*(V0 - V2) + 1);
            elseif LPV_opening(ii) || HPV_opening(ii)
                tetaLPVo_m(ii) = pi;
            end
            if  teta(ii) >= tetaLPVo_m(ii) || teta(ii) >= pi
                LPVb(ii) = 1;
            elseif change_31(ii) && teta(ii) >= tetaLPVo_switch(ii)
                LPVb(ii) = 1;
                change_31(ii) = 0;
            end
            if teta(ii) >= tetaLPVc_m || teta(ii) < alpha
                LPVb(ii) = 0;
            end

            %///2nd quadrant - CW motoring///
        elseif quadrant == 2

                %HPV
            if teta(ii) < alpha_cw && teta(ii) >= tetaLPVc_m
                HPVb(ii) = 0;
            end
            if teta(ii) >= alpha_cw || teta(ii) <= tetaHPVo_m && ~change_12(ii)
                HPVb(ii) = 1;
            elseif change_12(ii) && teta(ii) <= tetaHPVo_switch(ii)
                HPVb(ii) = 1;
            end
            if change_12(ii) && teta(ii) > pi
                change_12(ii) = 0;
            end

                %LPV
            if LPV_opening(ii)
                tetaLPVo_m(ii) = pi;
            elseif ~HPVb(ii) && ~HPV_closing(ii) && xH(ii) == 1
                V1 = V0 + Vd/2*(1 - cos(teta(ii) - ts*abs(tetadot)));
                V2 = exp(log(V1) + RHS_int);
                tetaLPVo_m(ii) = 2*pi - acos(2/Vd*(V0 - V2) + 1);
            end

            if teta(ii) <= tetaLPVo_m(ii) && teta(ii) >tetaLPVc_m && ~change_12(ii)
                LPVb(ii) = 1;
            end
            if teta(ii) <= tetaLPVc_m || teta(ii) > alpha_cw
                LPVb(ii) = 0;
            end
```

```
    %///3rd quadrant - CW pumping///
elseif quadrant == 3


        %LPV
    if teta(ii) <= tetaLPVc_p
        LPVb(ii) = 0;
    end
    if teta(ii) > tetaLPVc_p && teta(ii) <= tetaLPVo_p
        LPVb(ii) = 1;
    end

    if change_23(ii)
        LPVb(ii) = 0;
        if teta(ii) <= tetaLPVo_m(ii)
            LPVb(ii) = 1;
            change_23(ii) = 0;
        end
    end

        %HPV
    if HPV_closing(ii) && xH(ii) ~= 0
        tetaHPVo_p(ii) = 0;
    end
    if ~LPVb(ii) && ~LPV_closing(ii)  && xL(ii) == 1
        tetaLPVo_switch(ii) = tetaLPVc_p;
        V1 = V0 + Vd/2*(1 - cos(teta(ii) - ts*abs(tetadot)));
        V2 = exp(log(V1) - RHS_int);
        tetaHPVo_p(ii) = acos(2/Vd*(V0 - V2) + 1);
    end
    if teta(ii) > tetaHPVc_p && teta(ii) <= tetaHPVo_p(ii)
        HPVb(ii) = 1;
    elseif change_13(ii) && teta(ii) <= tetaHPVo_switch(ii)
        HPVb(ii) = 1;
        change_13(ii) = 0;
    end
    if LPVb(ii) && teta(ii) < pi || teta(ii) <= tetaHPVc_p || teta(ii) > alpha_max
        HPVb(ii) = 0;
    end

end

%%  --Special Conditions--

if IMPROVED && quadrant == 3 && quadrant_prev == 1 ...
        && ~HPVb(ii) && ~LPVb(ii) && teta(ii) < tetaLPVc_m
    change_13(ii) = 1;
end

if IMPROVED2 && quadrant == 1 && quadrant_prev == 3 ...
        && ~HPVb(ii) && ~LPVb(ii) && teta(ii) > tetaHPVc_p && teta(ii) < tetaLPVo_
    change_31(ii) = 1;
end

%---------------Lock mode:-----------------------------
```

```
if LOCKED
    if tetadot_ref == 0 && abs(e_tetadot) <= Vel_tolerance2 || stop
        HPVb(ii) = 0;
        LPVb(ii) = 0;
        if ~stop
            fprintf('\n STOPPED AT t = %.1f s \n', t);
        end
        stop=1;
        if tetadot_ref ~= 0
            stop = 0;
        end
    end
end % end LOCKED

%%       --- VALVE DYNAMICS ---

    %HPV opening
if HPVb(ii) && xH(ii) ~= 1 && ~HPV_closing(ii) ...
        || HPV_opening(ii)

    if ~HPV_opening(ii)
        HPV_opening(ii) = 1;    %start opening
    end

    xH(ii) = valve_dyn(poly_valve,t_HPV(ii));
    t_HPV(ii) = t_HPV(ii) + dt;

    if t_HPV(ii) >= ts + charge_time %actuator charge build-up
        t_HPV(ii) = ts;
        HPV_opening(ii) = 0;    %finish opening
    end
end

    %HPV closing
if HPVb(ii) == 0 && xH(ii) ~= 0 && HPV_opening(ii) == 0 ...
        || HPV_closing(ii) == 1

    if ~HPV_closing(ii)
        HPV_closing(ii) = 1;
    end

    xH(ii) = valve_dyn(poly_valve,t_HPV(ii));
    t_HPV(ii) = t_HPV(ii) - dt;

    if t_HPV(ii) <= - charge_time
        t_HPV(ii) = 0;
        HPV_closing(ii) = 0;
    end
end

    %LPV opening
if LPVb(ii) && xL(ii) ~= 1 && ~LPV_closing(ii) || LPV_opening(ii)

    if ~LPV_opening(ii)
        LPV_opening(ii) = 1;
    end
```

```
        xL(ii) = valve_dyn(poly_valve,t_LPV(ii));
        t_LPV(ii) = t_LPV(ii) + dt;

        if t_LPV(ii) >= ts + charge_time
            t_LPV(ii) = ts;
            LPV_opening(ii) = 0;
        end
    end

        %LPV closing
    if ~LPVb(ii) && xL(ii) ~= 0 && ~LPV_opening(ii) ...
            || LPV_closing(ii)

        if ~LPV_closing(ii)
            LPV_closing(ii) = 1;
        end

        xL(ii) = valve_dyn(poly_valve,t_LPV(ii));
        t_LPV(ii) = t_LPV(ii) - dt;

        if t_LPV(ii) <= - charge_time
            t_LPV(ii) = 0;
            LPV_closing(ii) = 0;
        end
    end

        %Plunger position limits:
    if xH(ii) > 1
        xH(ii) = 1;
    elseif xH(ii) < 0
        xH(ii) = 0;
    end
    if xL(ii) > 1
        xL(ii) = 1;
    elseif xL(ii) < 0
        xL(ii) = 0;
    end

    %---VALVE DYNAMICS END---
end

%% _______________MOTOR MODEL_________________

Vc = V0 + Vd/2*(1-cos(teta));                   %Cylinder displacement
Vcdot = Vd/2*sin(teta)*tetadot;                 %Rate of change cyl disp
xp = 0.5 + 0.5*cos(teta);                       %Generic piston motion

    %Variable bulk modulus:
epsilon_g = 1./ ( (1-epsilon_g0)/epsilon_g0 * ...
    ( p_atm./(p_atm + pc) ).^(-1/kappa) + 1 );
beta = 1./( (1/betaL) + epsilon_g./( p_atm + pc ) );

QH = xH*(1/kf).*sqrt(abs(pH-pc)).*sign(pH-pc);  %Flow high pressure valves
QL = xL*(1/kf).*sqrt(abs(pc-pL)).*sign(pc-pL);  %Flow low pressure valves
Tc = pc*Vd/2.*sin(teta);                        %Cylinder torques
```

```
pcdot = beta./Vc.*(QH-QL-Vcdot);                %Cylinder pressure gradients
Qin = sum(QH);                                  %Total flow in
Qout = sum(QL);                                 %Total flow out

%% _____________MECHANICAL MODEL______________


delta = r_eff*teta_shaft - (z_pl - z_pl_ini) + delta0;
deltadot = r_eff*tetadot - zdot_pl;

if delta <= 0
    Fw = 0;     %Wire strekk only valid for strekk
else
    Fw = delta*k_w + deltadot*b_w;
end

if loaded
    TL = (Fw + m_wout*(1-w_frac)*g-Fb_w*(1-w_frac))*r_eff + bb*tetadot;
else
    TL = -0.8*Tmax;
    if t >= t0 + ddt && t < t1 - ddt
        TL = -0.8*Tmax + 0.8*2*Tmax/(t1 - t0 - 2*ddt)*(t - (t0 + ddt));
    elseif t >= t1 - ddt
        TL = 0.8*Tmax;
    end
end

zdotdot_pl = (Fw + Fb_pl + Fb_w*w_frac ...
    - (m_pl + m_wout*w_frac)*g)/(m_pl + m_wout*w_frac);

Tm = sum(Tc);

tetadotdot = (Tm - TL)/Jeff;


%% _________________PLOTTING__________________

if plot_counter == interval
    plot_counter = 0;

    t_plot(counter) = t;
    tetadot_plot(counter) = tetadot*30/pi;
    Tm_plot(counter) = Tm;
    TL_plot(counter) = TL;
    pc_plot(:,counter) = pc*1e-5;
    xH_plot(:,counter) = xH;
    xL_plot(:,counter) = xL;
    QH_plot(:,counter) = QH*60000;
    QL_plot(:,counter) = QL*60000;
    teta_plot(counter) = teta_shaft;
    Vc_plot(:,counter) = Vc;
    Vcd_plot(:,counter) = Vcdot;
    teta_cyl(:,counter) = teta;
    alpha_plot(counter) = alpha;
    if quadrant == 2
        alpha_plot_qu(counter) = alpha_cw;
    else
        alpha_plot_qu(counter) = alpha;
    end
    xp_plot(:,counter) = xp;
    tetadotdot_plot(counter) = tetadotdot;
    Qin_plot(counter) = Qin;
    Qout_plot(counter) = Qout;
    e_tetadot_plot(counter) = e_tetadot;
    tetadot_ref_plot(counter) = tetadot_ref;
    tetadot_ctr_plot(counter) = tetadot_ref;
    E_tetadot_plot(counter) = E_tetadot;
    Tc_plot(:,counter) = Tc;
    alpha_max_plot(counter) = alpha_max;
    tetaLPVc_plot(counter) = tetaLPVc_m;
    HPVb_plot(:,counter) = HPVb;
    LPVb_plot(:,counter) = LPVb;
    HPV_opening_plot(:,counter) = HPV_opening;
    HPV_closing_plot(:,counter) = HPV_closing;
    LPV_opening_plot(:,counter) = LPV_opening;
    LPV_closing_plot(:,counter) = LPV_closing;
    quadrant_plot(counter) = quadrant;
    QIN(counter) = Vd/2*n_c*abs(tetadot*pi/30)*(1-cos(abs(alpha)))*2;
    tetaHPVo_p_plot(:,counter) = tetaHPVo_p;
    z_pl_plot(counter) = z_pl;
    zdot_pl_plot(counter) = zdot_pl;
    delta_plot(counter) = delta;
    deltadot_plot(:,counter) = deltadot;
    Fw_plot(counter) = Fw;

    counter = counter + 1;
end

%% _____________ITERATIONS & MISC_____________

%Motion
teta_shaft = teta_shaft + tetadot*dt;           %Shaft angle iteration
teta = mod(teta + tetadot*dt,2*pi);             %Individual cyl ang iteration
tetadot = tetadot + tetadotdot*dt;              %Ang vel iteration
z_pl = z_pl + zdot_pl*dt;
zdot_pl = zdot_pl + zdotdot_pl*dt;

%Chamber Pressure
pc = pc + pcdot*dt;                             %Cylinder pressure iteration

%Simulation
t = t + dt;                                     %Time increment
plot_counter = plot_counter + 1;                %Plotting
index = index + 1;

%Controller
E_tetadot = E_tetadot + e_tetadot*dt;           %Integral part
edot_tetadot = (e_tetadot_prev-e_tetadot)/dt;
e_tetadot_prev = e_tetadot;
if quadrant_prev ~= quadrant
    t_quad = t;
end
quadrant_prev = quadrant;
alpha_prev = alpha;

end

toc


%%  --pots--


for HIDE_ALL_PLOTS = 1:1

%% -- Strings and cylinder numbers --
if ~s1
    s1 = randi(n_c);
end

if ~s2
    s2 = randi(n_c);
end

if ~s3
    s3 = randi(n_c);
end

STR_S1 = num2str(s1);
STR_S2 = num2str(s2);
STR_S3 = num2str(s3);

FIG1 = ['Cylinder ' STR_S1];
FIG2 = ['Cylinder ' STR_S2];
FIG3 = ['Cylinder ' STR_S3];
PRE_VAL_STR_S1 = ['Pressure and Valves ' STR_S1];
PRE_VAL_STR_S2 = ['Pressure and Valves ' STR_S2];
PRE_VAL_STR_S3 = ['Pressure and Valves ' STR_S3];
FLOW_STR1 = ['Flows Cylinder ' STR_S1];
FLOW_STR2 = ['Flows Cylinder ' STR_S2];
FLOW_STR3 = ['Flows Cylinder ' STR_S3];

greeen = [0, 0.7, 0];
orankanskje = [255 129 0]/255;
dirty_cyan = [10 200 200]/255;
quad_line = 2*Tmax*[-1 1 -1]*0;
%     t_quadp = [t_quad-dt t_quad t_quad+dt];
t_quadp = [0 0 0];
quad_line_clr = dirty_cyan;

W = 17;
Figure_Size = [W 0.3*W];
Figure_PaperPosition = [0 0 Figure_Size];

Axis_Position = [1.7 1.1 Figure_Size(1)*0.885 0.75*Figure_Size(2)];
Axis_Font = 10;
lineW = 1.0;
close all

%%  PRESSURE AND VALVES


if BIGR_INFO_PLOTS
    %% CYL
    fig_cyl = figure('Name',FIG1,'NumberTitle','Off');
    hold on
    grid

    p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
    p2 = plot(t_plot,0.5*ones(1,length(t_plot)),'--','Color',[209,209,209]/255);
    p3 = plot(t_plot,pc_plot(s1,:)/pH/1e-5,'Color',[0 0, 0]/255,'LineWidth',lineW);
    p4 = plot(t_plot,teta_cyl(s1,:)/2/pi,'Color',[0, 0.7, 0 ],'LineWidth',lineW);
    p5 = plot(t_plot,0.99*xH_plot(s1,:),'--','Color','r','LineWidth',lineW);
    p6 = plot(t_plot,0.99*xL_plot(s1,:),'--','Color','b','LineWidth',lineW);

    hl = legend([p3 p4 p5 p6 ],'$\textnormal{p}_{\textnormal{c}}$','$\theta_c$','$\tex
        '$\textnormal{x}_L$');%,'$\pm\dot{\theta}_{ref}$','$\pm\dot{\theta}$',...
        %'$\alpha$');
    set(hl, 'Interpreter', 'latex');
    xtext = xlabel('$\textnormal{Time }[s]$');
    set(xtext, 'Interpreter', 'latex');
    ytext = ylabel('Normalised values');
    ytext.Interpreter = 'latex';
    ytext.Color = [1 1 1];
%         ylim([-0.01 1.2])
%         yticks(0:0.2:1);
%         xticks(0:2:26);

        %zoom in values: (sla ogsa av xticks
%         xlim([12.2 13.2])

%         fig_cyl.Units = 'Centimeter';
%         fig_cyl.PaperPosition = Figure_PaperPosition;
%         fig_cyl.Position = [0 17 Figure_Size];
%         ax = gca;
%         ax.Units = 'Centimeter';
%         ax.Position = Axis_Position;
%         ax.FontSize = Axis_Font;

%         ar = annotation('arrow');
%         ar2 = annotation('arrow');

    %%  FLOW
    fig_flow = figure('Name',FLOW_STR1,'NumberTitle','Off');
    grid
    hold on
    p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
    p2 = plot(t_plot,QH_plot(s1,:),'r','LineWidth',lineW);
    p3 = plot(t_plot,QL_plot(s1,:),'b','LineWidth',lineW);
    hl = legend([p2 p3],'Q$_H$','Q$_L$' );
    hl.Interpreter = 'latex';
    xtext = xlabel('$\textnormal{Time }[s]$');
    set(xtext, 'Interpreter', 'latex');
        ytext = ylabel('Flow $[l/min]$');
        ytext.Interpreter = 'latex';

            %normal values
%         xticks(0:2:26);
%         ylim([-100 50]) %cyl 21
%         yticks([-250:50:50])

            % peak values
%         ylim([-100 750])
%         yticks([-100:100:900])

            %zoom in values:
%         xlim([12.2 13.2])
%         ylim([-50 850])
%         yticks([0:200:850])


%         fig_flow.Units = 'Centimeter';
%         fig_flow.PaperPosition = Figure_PaperPosition;
%         fig_flow.Position = [0 9 Figure_Size];
%         ax = gca;
%         ax.Units = 'Centimeter';
%         ax.Position = Axis_Position;
%         ax.FontSize = Axis_Font;

        %% CYL2
%         fig_cyl2 = figure('Name',FIG2,'NumberTitle','Off');
%         hold on
%         grid
%
%         p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
%         p2 = plot(t_plot,0.5*ones(1,length(t_plot)),'--','Color',[209,209,209]/255);
%         p3 = plot(t_plot,pc_plot(s2,:)/pH/1e-5,'Color',[0 0, 0]/255,'LineWidth',lineW);
%         p4 = plot(t_plot,teta_cyl(s2,:)/2/pi,'Color',[0, 0.7, 0 ],'LineWidth',lineW);
%         p5 = plot(t_plot,0.99*xH_plot(s2,:),'--','Color','r','LineWidth',lineW);
%         p6 = plot(t_plot,0.99*xL_plot(s2,:),'--','Color','b','LineWidth',lineW);
%
%         hl = legend([p3 p4 p5 p6 ],'$\textnormal{p}_{\textnormal{c}}$','$\theta_c$','$\t
%             '$\textnormal{x}_L$');%,'$\pm\dot{\theta}_{ref}$','$\pm\dot{\theta}$',...
%             %'$\alpha$');
%         set(hl, 'Interpreter', 'latex');
%         xtext = xlabel('$\textnormal{Time }[s]$');
%         set(xtext, 'Interpreter', 'latex');
%         ytext = ylabel('Normalised values');
%         ytext.Interpreter = 'latex';
%         ytext.Color = [1 1 1];
%         xticks(0:2:26);
%
%         if IMPROVED3
%             ylim([-0.01 1.2])
%             yticks(0:0.2:1);
%         else
%             ylim([-0.01 1.2])
%             yticks(0:0.2:1);
%             if s2 == 38
%             ylim([-0.01 1.55]) %cyl 38
%             yticks(0:0.2:2.4);
%             end
%         end
%
% %         xlim([11.5 13])
%
%         fig_cyl2.Units = 'Centimeter';
%         fig_cyl2.PaperPosition = Figure_PaperPosition;
%         fig_cyl2.Position = [0 17 Figure_Size];
%         ax = gca;
%         ax.Units = 'Centimeter';
%         ax.Position = Axis_Position;
%         ax.FontSize = Axis_Font;

        %%  FLOW2
%         fig_flow2 = figure('Name',FLOW_STR2,'NumberTitle','Off');
%         grid
%         hold on
%         p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
%         p2 = plot(t_plot,QH_plot(s2,:),'r','LineWidth',lineW);
%         p3 = plot(t_plot,QL_plot(s2,:),'b','LineWidth',lineW);
%         hl = legend([p2 p3],'Q$_H$','Q$_L$' );
%         hl.Interpreter = 'latex';
%         xtext = xlabel('$\textnormal{Time }[s]$');
%         set(xtext, 'Interpreter', 'latex');
%         ytext = ylabel('Flow $[l/min]$');
%         ytext.Interpreter = 'latex';
%
%         ylim([-125 50])
%         yticks([-100:25:50])
%         xticks(0:2:26);
%         if IMPROVED4
%             ylim([-100 50])
%             yticks(-100:25:1000)
%         else
%             ylim([-100 750])
%             if s2 == 38
%                 ylim([-320 50]); %cyl 38?
%                 yticks(-300:50:900)
%             end
%         end
%
%
%         fig_flow2.Units = 'Centimeter';
%         fig_flow2.PaperPosition = Figure_PaperPosition;
%         fig_flow2.Position = [0 9 Figure_Size];
%         ax = gca;
%         ax.Units = 'Centimeter';
%         ax.Position = Axis_Position;
%         ax.FontSize = Axis_Font;
    %% OTHER CYLS

        %----------%

        figure('Name',FIG2,'NumberTitle','Off')
```

```
%         maximize

            subplot(2,1,1)
        hold on

        plot(t_plot,pc_plot(s2,:)/pH/1e-5,'Color',[99,99,99]/255,'LineWidth',0.5)%[255;140
        plot(t_plot,teta_cyl(s2,:)/2/pi,'Color',[0, 0.7, 0 ])

        plot(t_plot,xH_plot(s2,:),'--','Color','r','LineWidth',1)%'Color',[0.54, 0.17, 0.8
        plot(t_plot,xL_plot(s2,:),'--','Color','b','LineWidth',1)%'Color',[0,154,205]/255)

        plot(t_plot,sign(tetadot_ref_plot)/10+0.3,'Color',[128;0;128]/255)
        plot(t_plot,sign(tetadot_plot)/10+0.7,'Color',[139,90,0]/255)
        plot(t_plot,alpha_plot_qu/2/pi,':','LineWidth',1.05,'Color',[16,78,139]/255)
        plot(t_plot,0.5*ones(1,length(t_plot)),'--','Color',[209,209,209]/255)
        grid
        title(PRE_VAL_STR_S2)
        hl = legend('$\textnormal{p}_{\textnormal{c}}$','$\theta$','$\textnormal{x}_H$', .
            '$\textnormal{x}_L$','$\pm\dot{\theta}_{ref}$','$\pm\dot{\theta}$',...
            '$\alpha$');
        set(hl, 'Interpreter', 'latex');

        subplot(2,1,2)
        plot(t_plot,QH_plot(s2,:),'r',t_plot,QL_plot(s2,:),'b')
        grid
        title(FLOW_STR2)
        legend( 'Q_H', 'Q_L' )

        %----------%

        figure('Name',FIG3,'NumberTitle','Off')
%         maximize

            subplot(2,1,1)
        hold on

        plot(t_plot,pc_plot(s3,:)/pH/1e-5,'Color',[99,99,99]/255,'LineWidth',0.5)%[255;140
        plot(t_plot,teta_cyl(s3,:)/2/pi,'Color',[0, 0.7, 0 ])

        plot(t_plot,xH_plot(s3,:),'--','Color','r','LineWidth',1)%'Color',[0.54, 0.17, 0.8
        plot(t_plot,xL_plot(s3,:),'--','Color','b','LineWidth',1)%'Color',[0,154,205]/255)

        plot(t_plot,sign(tetadot_ref_plot)/10+0.3,'Color',[128;0;128]/255)
        plot(t_plot,sign(tetadot_plot)/10+0.7,'Color',[139,90,0]/255)
        plot(t_plot,alpha_plot_qu/2/pi,':','LineWidth',1.05,'Color',[16,78,139]/255)
        plot(t_plot,0.5*ones(1,length(t_plot)),'--','Color',[209,209,209]/255)
        grid
        title(PRE_VAL_STR_S3)
        hl = legend('$\textnormal{p}_{\textnormal{c}}$','$\theta$','$\textnormal{x}_H$', .
            '$\textnormal{x}_L$','$\pm\dot{\theta}_{ref}$','$\pm\dot{\theta}$',...
            '$\alpha$');
        set(hl, 'Interpreter', 'latex');

            subplot(2,1,2)
        title(FLOW_STR3)
        plot(t_plot,QH_plot(s3,:),'r',t_plot,QL_plot(s3,:),'b')
        grid
        title(FLOW_STR3)
        legend( 'Q_H', 'Q_L' )
    end

    %%  SHAFT ANGULAR VELOCITY

    % figure('Name','Shaft ','NumberTitle','Off')

    fig_shaft = figure('Name','Shaft ','NumberTitle','Off');
    hold on
    grid

    p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
    p2 = plot(t_plot,tetadot_ref_plot,'--','Color',orankanskje,'LineWidth',lineW*2);
    p3 = plot(t_plot,tetadot_plot*pi/30,'b','LineWidth',lineW);
    hl = legend([p2 p3],'$\dot{\theta}_{\textnormal{ref}}$','$\dot{\theta}_m$');
    hl.Interpreter = 'latex';
    ytext = ylabel('Angular Velocity $[rad/s]$');
    ytext.Interpreter = 'latex';
    xtext = xlabel('$\textnormal{Time }[s]$');
    xtext.Interpreter = 'latex';
%     ylim([-1.5 -0.5])
%     yticks([-1.5:0.1: -0.5])
%     xticks(0:2:26);
%
%     fig_shaft.Units = 'Centimeter';
%     fig_shaft.PaperPosition = Figure_PaperPosition;
%     fig_shaft.Position = [0 1 Figure_Size];
%     ax = gca;
%     ax.Units = 'Centimeter';
%     ax.Position = Axis_Position;
%     ax.FontSize = Axis_Font;


    %%  TORQUE

    Tmax_plot = Tmax*ones(1,length(t_plot));

    fig_torque = figure('Name','Torque','NumberTitle','Off');
    hold on
    grid


    p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
    p2 = plot(t_plot,Tm_plot/1e3,'b','LineWidth',lineW);
    p3 = plot(t_plot,Tmax_plot/1e3,'r--','LineWidth',lineW);
    p4 = plot(t_plot,TL_plot/1e3,'LineWidth',lineW,'Color',orankanskje);

    hl = legend([p2 p4 p3],'T$_m$','T$_L$','T$_{max}$');
    hl.Interpreter = 'latex';
    ytext = ylabel('Torque  $[kNm]$');
    ytext.Interpreter = 'latex';
    xtext = xlabel('$\textnormal{Time }[s]$');
    xtext.Interpreter = 'latex';
%     xticks(0:2:26);
%     if IMPROVED
%         ylim([0.9*min(TL_plot) 1.1*Tmax]*1e-3)
%     else
%         ylim([200 350])
%         yticks([200:50:350])
%     end
%     ylim([-300 350])
%     yticks([-300:100:350])


%     fig_torque.Units = 'Centimeter';
%     fig_torque.PaperPosition = Figure_PaperPosition;
%     fig_torque.Position = [W+5 17 Figure_Size];
%     ax = gca;
%     ax.Units = 'Centimeter';
%     ax.Position = Axis_Position;
%     ax.FontSize = Axis_Font;


    %%  ALPHA

    fig_alpha = figure('Name','Alpha','NumberTitle','Off');
    grid on
    hold on

    p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
    p2 = plot(t_plot,alpha_plot,'b','LineWidth',lineW);

    hl = legend([p2],'$\alpha$');
    set(hl, 'Interpreter', 'latex');
    ytext = ylabel('Angle $[rad]$');
    ytext.Interpreter = 'latex';
    xtext = xlabel('$\textnormal{Time }[s]$');
    xtext.Interpreter = 'latex';
%     xticks(0:2:26);
%     ylim([-2.3 2.3])
%     yticks([-2.5:0.5:2.4])
%     if IMPROVED
%         ylim([0.99*min(alpha_plot) 1.01*max(alpha_plot)])
%         yticks(2:0.05:2.3);
%     else
%         ylim([1.8 2.3])
%         yticks([1.8:0.1:2.3])
%     end


%     fig_alpha.Units = 'Centimeter';
%     fig_alpha.PaperPosition = Figure_PaperPosition;
%     fig_alpha.Position = [W+5 9 Figure_Size];
%     ax = gca;
%     ax.Units = 'Centimeter';
%     ax.Position = Axis_Position;
%     ax.FontSize = Axis_Font;

    %% Vc_plot
    % figure
    % plot(t_plot,Vc_plot(s1,:)*1e6,'b')
    % grid
    % legend('V_c')

    %% Rate of change cylinder volume

    % figure('Name','Rate of change cylinder volume','NumberTitle','Off')
    % plot(t_plot,Vcd_plot(s1,:)*60000,'b')
    % grid
    % hl = legend('$\dot{V}_c$');
    % set(hl, 'Interpreter', 'latex');

    %% Teta cyl

    % figure
    % plot(t_plot,teta_cyl(s1,:))
    % grid
    % legend('\theta_{cyl}')

    %% Ang acc

    % figure('Name','Acc Ang','NumberTitle','Off')
    % plot(t_plot,tetadotdot_plot,'b')
    % grid
    % hl = legend('$\ddot{\theta}$');
    % set(hl, 'Interpreter', 'latex');

    %% Flows

%     fig_flows = figure('Name','Flows','NumberTitle','Off');
%     hold on
%     grid
%
%
%     p1 = plot(t_quadp,quad_line,'Color',quad_line_clr,'LineWidth',0.7*lineW);
%     p2 = plot(t_plot,Qin_plot*60000,'r','LineWidth',lineW);
%     p3 = plot(t_plot,Qout_plot*60000,'b','LineWidth',lineW);
%
%     hl = legend([p2 p3],'Q$_{in}$','Q$_{out}$');
%     set(hl, 'Interpreter', 'latex');
%     ytext = ylabel('Flow $[l/min]$');
%     ytext.Interpreter = 'latex';
%     xtext = xlabel('$\textnormal{Time }[s]$');
%     xtext.Interpreter = 'latex';
%     xticks(0:2:26);
%     if IMPROVED4
%         ylim([-750 750])
%         yticks(-1000:500:2500);
%     else
%         ylim([-1000 1000])
%         yticks(-1000:500:2500)
%     end
%
%
%     fig_flows.Units = 'Centimeter';
%     fig_flows.PaperPosition = Figure_PaperPosition;
%     fig_flows.Position = [W+5 1 Figure_Size];
%     ax = gca;
%     ax.Units = 'Centimeter';
%     ax.Position = Axis_Position;
%     ax.FontSize = Axis_Font;

    %% Flow inn

    figure('Name','Flow inn','NumberTitle','Off')
    for ii = 1:n_c
        hold on
        plot(t_plot,QH_plot(ii,:))
    end
    plot(t_plot,Qin_plot*60000,'r')
    grid

    %% Flow ud

    figure('Name','Flow ud','NumberTitle','Off')
    for ii = 1:n_c
        hold on
        plot(t_plot,QL_plot(ii,:))
    end
    plot(t_plot,Qout_plot*60000,'r')
    grid

    %% ctrl error

%     figure('Name','Controller Error','NumberTitle','Off')
%     plot(t_plot,Kp_vel*e_tetadot_plot,t_plot,Kp_vel*E_tetadot_plot/Ti_vel)
%     grid
%     hl = legend('$\textnormal{K}_{\textnormal{p}}\cdot\textnormal{e}_{\dot{\theta}}$',..
%         '$^{\textnormal{K}_{\textnormal{p}}}/_{\textnormal{T}_{\textnormal{i}}}\cdot\int
%     set(hl, 'Interpreter', 'latex');

    %% Torque cylinders

%     figure('Name','Torqueses','NumberTitle','Off')
%     for ii = 1:n_c
%         hold on
%         plot(t_plot,Tc_plot(ii,:)/1e3)
%     end
%     plot(t_plot,Tm_plot/1e3,'r')
%     grid

    %% Vc vs QH

    % figure
    % plot(t_plot,QH_plot(s,:),'b',t_plot,QL_plot(s,:),'g',t_plot,abs(Vcd_plot(s,:))*60000
    %
    % diffH = QH_plot(s,:)/60000-abs(Vcd_plot(s,:));
    % diffL = QL_plot(s,:)/60000-abs(Vcd_plot(s,:));
    %
    % figure
    % plot(t_plot,diffH,'b',t_plot,diffL,'r')
    %% LPVc

    % figure
    %
    % plot(t_plot,alpha_max_plot,'r',t_plot,tetaLPVc_plot,'b')

    %%

    ss = s3;

    if ss == s1
        BOOL_STR = STR_S1;
    elseif ss == s2
        BOOL_STR = STR_S2;
    elseif ss == s3
        BOOL_STR = STR_S3;
    end

%     FIG_HPV= ['HPV cyl ' BOOL_STR];
%     figure('Name',FIG_HPV,'NumberTitle','Off')
%
%     plot(t_plot,HPVb_plot(ss,:),t_plot,xH_plot(ss,:), ...
%         t_plot,HPV_opening_plot(ss,:)/2,t_plot,HPV_closing_plot(ss,:)/4)
%     grid
%     legend('HPVb','x_H','HPV_opening','HPV_closing')
%     ylim([-0.1 1.1])
%
%         %%
%
%     FIG_LPV= ['LPV cyl ' BOOL_STR];
%     figure('Name',FIG_LPV,'NumberTitle','Off')
%
%     plot(t_plot,LPVb_plot(ss,:),t_plot,xL_plot(ss,:), ...
%         t_plot,LPV_opening_plot(ss,:)/2,t_plot,LPV_closing_plot(ss,:)/4)
%     grid
%     legend('LPVb','x_L','LPV_opening','LPV_closing')
%     ylim([-0.1 1.1])


    %%  Quadrant

%     figure('Name','Quadrant','NumberTitle','Off')
%     grid
%     plot(t_plot,quadrant_plot)
%     ylim([0.95 4.05])
%     xlim([-0.05 T+0.05])

    %% tetaHPVo_p

%     figure
%     grid
%     plot(t_plot,tetaHPVo_p_plot(s3,:))

    %% error
```

```
%     figure('Name','Error Vel','NumberTitle','Off')
%     grid
%     plot(t_plot,e_tetadot_plot)

    %%  --Pressures

        figure('Name','Pressures','NumberTitle','Off')
    for ii = 1:n_c
        hold on
        plot(t_plot,pc_plot(ii,:))
    end
    grid
 end
```

A.3.1 Function: Find Alpha FF

```
function alpha_ff = find_alpha_ff(TL,coeff)
%Approximate alpha_ff from polyfit

%coeff(1)*TL^10

alpha_ff = coeff(1)*TL^10 + coeff(2)*TL^9 + coeff(3)*TL^8 + ...
    coeff(4)*TL^7 + coeff(5)*TL^6 + coeff(6)*TL^5 + coeff(7)*TL^4 + ...
    coeff(8)*TL^3 + coeff(9)*TL^2 + coeff(10)*TL^1 + coeff(11)*TL^0;


% alpha_ff = coeff(11)*TL^10 + coeff(10)*TL^9 + coeff(9)*TL^8 + ...
%     coeff(8)*TL^7 + coeff(7)*TL^8 + coeff(6)*TL^5 + coeff(5)*TL^4 + ...
%     coeff(4)*TL^3 + coeff(3)*TL^2 + coeff(2)*TL^1 + coeff(1)*TL^0;
end
```

A.3.2 Function: Int Var Beta

```
function [RHS_int] = int_var_beta(pc_from,pc_to,betaL,epsilon_g0,p_atm,kappa)
%Returns the RHS of the integral descending from the continuity equation
%   Computed with variable beta by numerical integral (forward Euler)

pc = pc_from;
dpc = 5;
RHS_int = 0;

if pc_from > pc_to

    while pc >= pc_to

            %bulk modulus:
        epsilon_g = 1./ ( (1-epsilon_g0)/epsilon_g0 * ...
            ( p_atm./(p_atm + pc) ).^(-1/kappa) + 1 );
        beta = 1./( (1/betaL) + epsilon_g./( p_atm + pc ) );
            %RHS integral:
        RHS_int = RHS_int + 1/beta*dpc;
        pc = pc - dpc;

    end
elseif pc_to > pc_from

    while pc <= pc_to

            %bulk modulus:
        epsilon_g = 1./ ( (1-epsilon_g0)/epsilon_g0 * ...
            ( p_atm./(p_atm + pc) ).^(-1/kappa) + 1 );
        beta = 1./( (1/betaL) + epsilon_g./( p_atm + pc ) );
            %RHS integral:
        RHS_int = RHS_int + 1/beta*dpc;
        pc = pc + dpc;

    end

end
```

A.3.3 Function: Valve Dyn

```
function [x] = valve_dyn(a,t)
%Valve motion profile based on polynomial approximation of valve dynamics
%   Takes in a coefficient vector "a" and  a time value "t" and outputs
%a position "x"

x = a(1)*t^10 + a(2)*t^9 + ...
                a(3)*t^8 + a(4)*t^7 + a(5)*t^6 ...
                + a(6)*t^5 + a(7)*t^4 + a(8)*t^3 ...
                + a(9)*t^2 + a(10)*t^1 + a(11)*t^0;

end
```

A.3.4 Function: Vel Profile

```
function [ vel t ] = motion( q,time,dt,T )

% Calculates a motion profile based on required specifications for start
% and end positions, time interval and time step.

t0=time(1,1);   %start time
t1=time(2,1);   %end time
t = linspace(t0,t1,T/dt);

%q=T*a
T=zeros(4);
T(1,1:4)=[t0^3 t0^2 t0 1];      %Start velocitiy
T(2,1:3)=[3*t0^2 2*t0 1];       %Start acceleration
T(3,1:4)=[t1^3 t1^2 t1 1];      %End velocitiy
T(4,1:3)=[3*t1^2 2*t1 1];       %End acceleration
```

```
a=T\q;           %polynomial coefficients

L=length(t);

%for reference:
% a0=a(4);
% a1=a(3);
% a2=a(2);
% a3=a(1);

%The equations are repeated for each time step:
vel = ones(1,L).*a(4) + t.*a(3) + t.^2.*a(2) + t.^3.*a(1);
end
```