Design of a compensation algorithm to maximize the smoothness of a hydraulic motion system

Design of a compensation algorithm to maximize the smoothness of a hydraulic motion system

R.J.M. de Jong

DCT 2005.79

Traineeship committee:
Dr.Ir. M.J.G. van de Molengraft *
Ir. P.Piatkiewitz **

* Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology

** Rexroth-Hydraudyne Boxtel
Systems and Engineering
Department Design and Development

Eindhoven, June 2005

Preface

This report contains the results of a sixteen week traineeship which has been carried out at the Design and Development section, which is part of System and Engineering at Rexroth-Hydraudyne in Boxtel, the Netherlands. I would like to thank all the members of the Design and Development group, especially Philippe Piatkiewitz and Coos Conijn for their support during this traineeship. I also would like to thank René van de Molengraft for his advices during our discussions.

Abstract

6-DOF platform motion systems, known as hexapods, are widely used for simulation-purposes. For example, these simulators are used for education like in flightsimulators for pilot and cabin-crew training as well as in the entertainment industry (IndyCar simulators and virtual reality). The sense of reality by the driver is the most important aspect in these kind of simulators, so excessive disturbances are not allowed. Some of the disturbances that humans are most susceptible to are small bumps in the motion system, causing accelerations that will be noticed by the driver.

When an actuator changes its direction of motion a disturbance of that kind can occur, which is known as the 'reversal bump'. The main cause lies in the non-linearity around the point where the valve is totally closed when this valve is part of a hydraulic motion system. In the current control software as used at Rexroth-Hydraudyne a compensation function is available, which needs to be adjusted by hand before system operation. However, the behavior of the valve can change under the influence of temperature and it relies on the usage of the motion system. So, there is a demand for a compensation algorithm that adjusts the compensation during system operation.

To visualize the main effects of the non-linearity in the valve on the actuator, a simulation is made of a hydraulic actuator. The valve-dynamics, flow out of the valve, continuity and compressibility of the oil and the forces acting on the actuator are the most important components. To create a Simulink-model in which the smoothness problem occurs, the smallest motion system of the Rexroth-Hydraudyne product-range (known as Micro Motion System) is used to perform the experiments. Apart from the available data, the main results are given regarding friction and smoothness, leading to assumptions for the eventual model. Results of the simulation are then compared to the experimental results, leading to an eventual model that can be used as a starting point to find a compensation algorithm.

In the so-called 'flow-curve', the non-linearity that is present in the valve can be visualized. This flow-curve establishes the relation between valve setpoint and piston velocity. The goal of the compensation is to cancel the non-linearity and thus make the relation between valve setpoint and piston velocity a linear one. As a result of this, the peaks in the acceleration profile will disappear. Because of the fact that the piston velocity has to be determined out of the differentiated position feedback-signal (which is available in the control-loop), a low-pass filter must be used in order ro reduce the effect of the measurement noise. This, in turn, causes phase-loss in the velocity-signal. Therefore, the decision has been made to use the difference between the actual relation between valve setpoint and piston velocity and the desired one to update the parameters of an algebraic compensation function. By doing so, the phase is less important and buffers can be used to store the actual relation over a certain amount of time.

First of all, the use of an iterative approach is investigated that adjusts itself until the desired linear relation is reached. Two different algebraic compensation functions have been considered for this purpose and the differences in simulation results have been discussed. The choice of compensation function seemed to be of great importance to be able to use an iterative compensation technique. When the compensation function is not able to provide the required compensation voltage (especially at the time when the valve is totally closed) because of its shape, the iterative approach may not be the right choice. This problem has been analyzed and solved by using actual differences instead of the

iterative process. The final approach that has been investigated is based on the use of buffers to store the relation between the valve setpoint and piston velocity. By storing this relation over a user definable amount of time, parameters can be updated before a new storage starts. Also a weighting function has been applied to the buffers to be able to base the parameters for the most on recent information. The results show that this is the safest option to create a self-adjusting compensation function.

Samenvatting

Bewegingssystemen met 6 graden van vrijheid, zogenaamde hexapods, worden veel gebruikt voor simulatie-doeleinden. Hierbij kan men denken aan bijvoorbeeld vlieg- en racewagensimulators, zowel voor gebruik in educatieve toepassingen (pilootopleiding, cabin-crew training) als voor entertainment-doeleinden (IndyCar-racespel, virtual reality). Een belangrijk aspect hierbij is dat het in de beleving van de piloot of bestuurder de werkelijkheid zeer goed moet kunnen nabootsen. Hierbij zijn verstoringen zoals kleine tikjes in het bewegingssysteem dus niet geoorloofd. Dit zijn verstoringen waar een mens zeer gevoelig voor is en deze leiden tot ongewenste versnellingen bij de bestuurder.

Wanneer een actuator van richting veranderd kan een dergelijke verstoring optreden, welke bekend staat als de zogenaamde 'omkeerbump'. De belangrijkste oorzaak hiervan is de niet-lineariteit in de nul-doorgang van de klep in een hydraulisch bewegingssysteem. In de huidige besturingssoftware die bij Rexroth-Hydraudyne wordt gebruikt voor de hydraulische systemen is een compenserende functie aanwezig, welke bij het in bedrijf stellen handmatig moet worden afgesteld. Het gedrag van de klep kan echter als functie van de temperatuur varieren en is tevens afhankelijk van de mate van gebruik van het systeem. Er is dus behoefte aan een compensatie-algoritme dat zichzelf instelt en aanpast gedurende het in bedrijf zijn van het bewegingssysteem.

Om de effecten van de niet-lineariteit van de klep in kaart te brengen is er een simulatie gemaakt van een hydraulische actuator. De belangrijkste onderdelen hierbij zijn: de klepdynamica, de stroming uit de klep, massabehoud en compressibiliteit van de olie en de krachten werkend op de zuiger. Om te komen tot een Simulink-model waarin de smoothness-problemen kunnen worden gevisualiseerd is het kleinste bewegingssysteem uit de serie bij Rexroth-Hydraudyne gebruikt om experimenten op uit te voeren. Naast de data die voorhanden was, zijn de werkelijke resultaten in kaart gebracht omtrent wrijving en smoothness en zijn er diverse aannamen voor het uiteindelijke model gemaakt. De resultaten van de simulatie zijn vergeleken met de resultaten uit het experiment en er is gekeken of het model bruikbaar is om als uitgangspunt te dienen voor het implementeren van een compensatie-functie.

De niet-lineariteit van de klep uit zich in de zogenaamde 'flow-curve', welke de relatie vastlegt tussen het klepsetpoint en de snelheid van de zuiger. Het doel van de compensatie is om dit verband lineair te maken, dit heeft namelijk als gevolg dat de pieken in het versnellingsprofiel verdwijnen. Vanwege het feit dat de snelheid moet worden verkregen door differentiatie van het positie-feedbacksignaal (in de regellus beschikbaar), zal door de aanwezigheid van meetruis een low-pass filter moeten worden gebruikt, wat tevens weer fase-verlies met zich meebrengt. Er is dan ook voor gekozen om het verschil tussen het klepsetpoint en de snelheid te gebruiken om parameters van een gladde compensatie functie aan te passen. Dit heeft als voordeel dat de fase niet van belang is en dat buffers kunnen worden gebruikt om de relatie over een bepaalde tijd op te slaan.

Als eerste mogelijkheid is een iteratieve functie onderzocht die zichzelf update net zolang totdat de gewenste relatie tussen het klepsetpoint en de snelheid van de zuiger lineair is. Hiervoor zijn twee mogelijke algebraische compensatiefuncties gebruikt en de resultaten van de simulatie zijn met elkaar vergeleken. Uit deze resultaten blijkt meteen het belang van de keuze van compensatiefunctie om de niet-lineariteit te kunnen opheffen. Is zo'n functie niet in staat om de vereiste compensatie te leveren (vooral door zijn vorm rond de oorsprong), dan is het gebruik van een iteratieve methode wellicht

niet te prefereren. Als laatste optie is het gebruik van buffers bekeken, uitgaande van de actuele relatie tussen klepsetpoint en zuigersnelheid. Door gedurende een in te stellen tijd deze relatie vast te leggen, kunnen eens per periode de parameters worden bepaald, waarna een nieuwe meting kan beginnen. Door tevens met een weegfunctie over deze buffer te werken, is het mogelijk zowel de informatie uit het verleden als de meest recente informatie (hier ligt dan de nadruk op) te gebruiken. Uit de resultaten blijkt dat dit de veiligste optie is om tot een zichzelf instellende en aanpassende compensatiefunctie te komen.

Contents

Chapter 1

Introduction

1.1 Motivation

Smoothness is probably the most important topic in the performance of motion systems that are used for simulation purposes. Unwanted accelerations will be noticed by the driver and as a consequence the sense of reality will decrease considerably, which is known as a 'false cue'. In hydraulic systems the non-linearity in the valve is the main cause for smoothness problems. This non-linearity can be assigned to the overlap in the valve, causing dead-band behavior around the mid-position of the valve spool. This influences the flow out of the valve, which can be seen in the flow-curve, and therefore the piston velocity (which is proportional to the flow). In the flow-curve the dead-band behavior can be recognized, so a non-linearity will be present in the velocity profile, causing peaks in the acceleration profile.

To improve the smoothness of the hydraulic motion system a compensation is required to cancel the valve non-linearity. The main goal is to force the non-linear relation between the valve setpoint and the piston velocity into a linear one. The valve characteristics vary with time and temperature and therefore the compensation should also be adjusted when the flow-curve changes while the system is in operation.

So, in order to visualize the smoothness of a hydraulic motion system a simulation is required. Therefore, the main parts of the hydraulic actuator have to be modeled. The Micro Motion System will be used to perform measurements to indicate the smoothness problem and provide numerical data for use in the simulation model.

In this report a compensation algorithm will be derived that cancels the non-linearity for the static case, which will be extended to a self-adjusting compensation. The next section presents the results of a literature study. After that the problem statement will be given, followed by the outline of this report.

1.2 Literature review

A hydraulic servo-system is an arrangement of individual components, interconnected to provide a desired form of hydraulic drive. The basic structure of a hydraulic system consists of a hydraulic power supply (pump driven by an electric motor), a control element (valve), an actuating element (cylinder) and other elements such as pipelines and measuring devices. The pump converts the available mechanical power from the prime mover (the electric motor) to hydraulic power at the actuator. A valve is used to control the direction of oil flow, the level of power produced and the amount of fluid and pressure to the actuator. The linear actuator, being the cylinder, converts the hydraulic power to usable mechanical power output at the point required. The hydraulic oil provides direct transmission and

control, as well as lubrication of components. Fluid storage (tank and accumulator) and conditioning equipment ensure sufficient quality, quantity and cooling of the oil. Modeling of these main parts of the hydraulic drive is described in [8] and [2]. Viersma [8] proposes a model for an asymmetric motor controlled by a three-way valve. More in-depth treatment can be found in the book by Jelali and Kroll [2], where all the basic elements are described and models are derived for each component. Also a distinction is made between experimentally and physically based modeling. The friction identification that will be used in this report as well as the basic idea in compensation for dead-band behavior is also presented in this book. The lecture notes by Viersma and Teerhuis [9] as well as the book by Stuyvenberg and Vinke [7] have been used in order to understand the technical drawings of the hydraulic drive with definitions of all the different symbols as described in the Micro Motion System manual by Rexroth-Hydraudyne [5]. Definitions on smooth mathematical functions can be found in [3].

1.3 Problem statement

Hydraulic motion systems used for simulation vehicles require a great sense of reality. Therefore, unwanted accelerations caused by non-linearity in the valve are not allowed. Because of the fact that the valve characteristics depend on temperature, a compensation is required that adjusts itself once the system is moving. So, the flow-curve needs to be evaluated continuously to calculate the required compensation in order to cancel the non-linearity and improve the smoothness of the hydraulic motion system. This leads to the following problem statement:

Design an adaptive compensation algorithm to maximize the smoothness of a hydraulic motion system.

1.4 Outline of the report

Chapter 2 presents the setup of the Micro Motion system on which the experiments will be performed. All the different subsystems will be described as well as the control loop for the hydraulic actuators as it is available in the software. The smoothness problem will be visualized and a friction model will be derived.

In chapter 3 the main parts of the hydraulic drive will be modeled according to physical relations. A Simulink-model will be created to perform numerical simulations. Results of the simulation will be shown, indicating the numerical problems that may occur and solving these to gain a model that acts as a starting point for the design of a compensation.

Modeling of the hydraulic system is based on several assumptions and recommendations. Therefore, the results of the simulation will be compared to the experimental results in chapter 4, indicating the main differences.

The design of a self-adjusting control algorithm will be treated in chapter 5. The basic idea behind the control algorithm will be given as well as some approaches to create a compensation algorithm, indicating the problems and advantages in the different methods. This will lead to conclusions and recommendations that will be given in chapter 6.

Chapter 2

Presentation of the Micro Motion System

2.1 Introduction

This chapter presents the setup of the Micro Motion System. The different subsystems will be discussed as well as the operation of the complete system. Also measurements will be performed to indicate the smoothness of the Micro Motion System and to identify the friction forces.

2.2 Description of the Micro Motion System

The Micro Motion System is a 6-DOF motion system consisting of a motion base assembly, a hydraulic power unit and an electronic control system. The electronic control system has three main parts, namely the Power Unit Control Cabinet, the Motion Control Cabinet and the Motion Computer. Optionally a host computer can be connected to the motion computer.

2.2.1 Motion base assembly

The motion base consists of a base frame that can be bolted to the floor. Six joints attached to this base frame (equipped with roller bearings to assure clearance-free rotation) are connected to the hydraulic actuators. One end of the hydraulic actuators is connected to the joints on the base frame, the other end is connected to the joints on the moving platform. On top of this platform a payload can be mounted with a maximal weight of approximately 600 kg.

Figure 2.1 shows the 6-DOF motion system. The schematic representation (left-hand picture) shows the axis definition according to the main directions (minimal set of generalized coordinates). The main directions are X,Y,Z, and the rotations around these axes and are also known as surge, sway, heave, roll pitch and yaw, respectively. In the right-hand picture the motion base, the six hydraulic actuators and the motion platform can be recognized.

Hydraulic actuator

As can be seen in (Fig. 2.1) the Micro Motion System is equipped with six hydraulic actuators. A hydraulic actuator is a linear actuator to convert hydraulic energy provided by a pump and processed by the control element (valve) into useful work (and consequently into power and mechanical energy

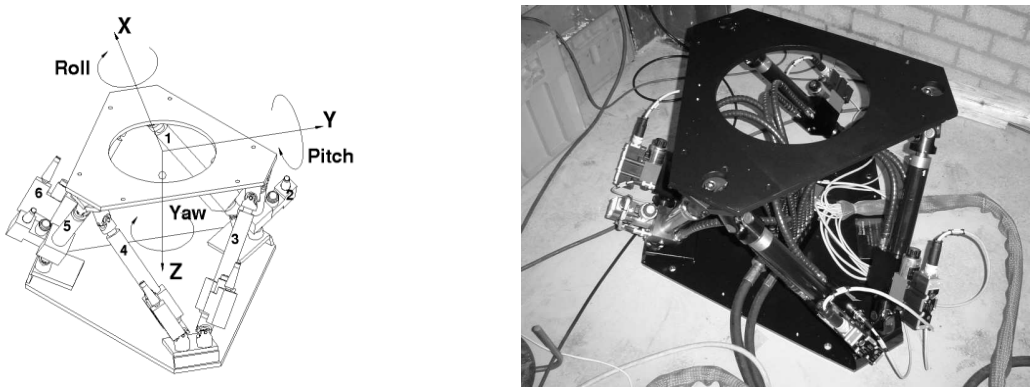


Figure 2.1: motion base assembly (schematic and reality) [Rexroth-Hydraudyne]

respectively). In this system an asymmetric, double-acting cylinder is used. The cylinder is called asymmetric, because the voids to be filled with oil are unequal for extension and retraction. The double-acting property of the actuator allows hydraulic force to be applied in both directions. To measure the position of the piston (the total stroke is 200 mm) a displacement transducer is built-in at the bottom-side of the cylinder. Soft end stops are incorporated in the design of the actuator to prevent peak or shock loads on the payload or passengers in case of control- or operator failures.

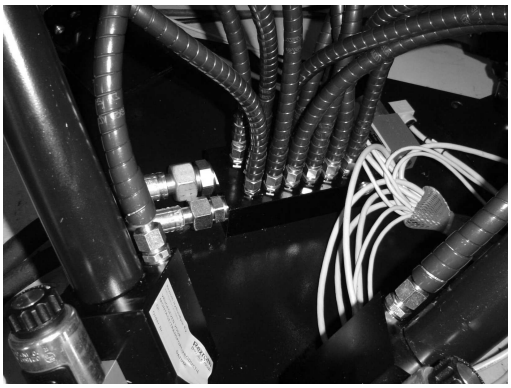


Figure 2.2: valve manifold [Rexroth-Hydraudyne]

A central valve manifold (see Fig. 2.2) distributes the hydraulic power to each of the motion actuators. To control the oil-flow to and from the bottom side of each cylinder a three-way proportional valve is used. The valve is named proportional, because of the fact that the output variable (the valve spool position) is proportional to the the input signal (input voltage). The fundamental principle of operation of the proportional valve is based on the use of a proportional solenoid (the electrical operator), which moves the valve spool to the desired position. The stroke of the valve spool is approximately 1 mm. A valve can be classified by:

- The number of 'ways' or ports flow can enter and leave the valve. The valve requires a supply (indicated by P, the pressure line), a return (indicated by T, the tank line) and at least one line (A) to the load. In this case a three-way valve is used, so a P,T and A-port is available.
- The number of switching positions. This can best be clarified by looking at the symbol that is used for this valve, see (Fig. 2.3). The valve is known as a 3/3-valve, so a three way valve (P,T and A) with three switching positions: from load to tank (A-T), from pressure to load (P-A) and a position where both the tank and pressure line are connected to the A-line.

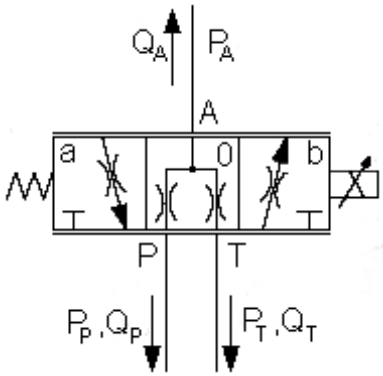


Figure 2.3: 3/3-valve

- The method of valve actuation to move into an alternative position. In this case the external signal command is electrical and the internal signal command is a spring force.
- The center type of the valve, which can be either open, closed or critical when the valve is in its neutral position. This valve, like most of the proportional valves, has overlap at middle position, causing a non-smooth flow-signal curve.

2.2.2 Hydraulic Power Unit (HPU)

To provide the hydraulic power to the actuators a special unit, known as Hydraulic Power Unit (HPU) is available (see Fig. 2.4). The HPU consists of an electric motor (2.5 kW) that drives the variable vane pump, which is located in the oil reservoir that has a capacity of approximately 50 liters. The choice of a variable vane pump is based on the fact that it only delivers oil flow when and as required by the system. Also an accumulator is present on the HPU to temporarily provide extra oil supply when required. The oil that is used in the system is hydraulic mineral oil, known as Shell Tellus T46. The system pressure is approximately 70 bar.

Figure 2.4: hydraulic power unit [Rexroth-Hydraudyne]

2.2.3 Electronic control system

The electronic control system mainly consists of three components, namely the Motion Computer (MC), the Motion Control Cabinet (MCC) and the Power Unit Control Cabinet (PUCC). The MC contains the D/A, A/D and digital I/O boards to control the state of the motion base. It is also possible to use a host computer that communicates with the motion computer via a ethernet interface. In the software the state of the motion base and the HPU can be evaluated. Also an axis transformation from 'degrees of freedom data' from the host computer to the six individual setpoint signals can be performed. On the other hand, an inverse axis transformation is provided from the six position feedback signals to the actual platform position in degree of freedom for use in fixed screen or projection systems. Feedback signals from the built-in position sensors are connected to the conditioning electronics in the MCC to be able to use them in the position control loop. The PUCC contains the power distribution system, the motor starter relay and fuses for the HPU and the connecting cables to the MCC. Figure 2.5 shows the setup of the Micro Motion System, the different parts and connections are also indicated.

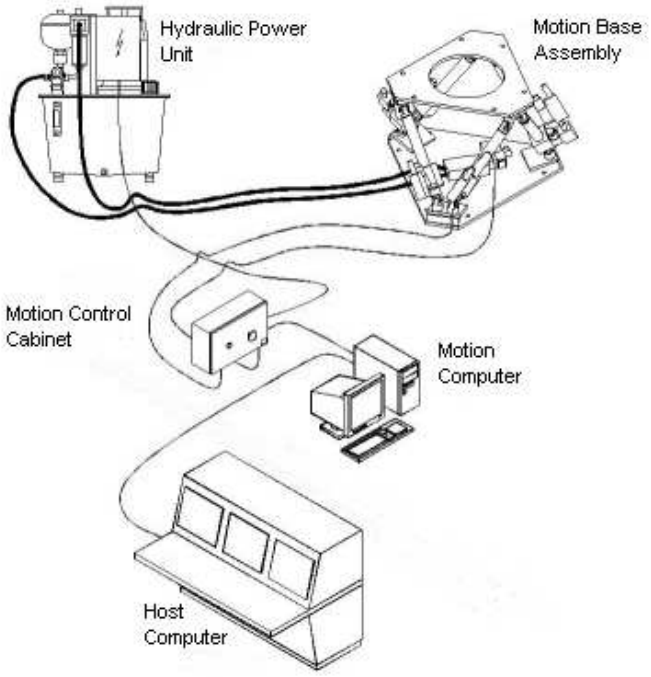


Figure 2.5: setup of the Micro Motion System [Rexroth-Hydraudyne]

2.2.4 Control software

On the motion computer DOS-based software is installed to control the complete system, running with a sampling frequency of 500 Hz. The system can be used in two different modes, namely normal mode for operations commanded by the host computer and manual mode for operations using the built-in signal generator. This signal generator offers the possibility to generate reference signals of several types, such as a sine wave, a block signal or a user specified input (loaded from a file). In the main window operation of the system in DOF-mode or in actuator-mode can be selected, by choosing actuator-mode the different actuators can be driven separately. From the main window it is

also possible to check the state of the motion base and possible error messages. The pump can be started and the system can be driven to the neutral state, which is the state of the system when all the actuators are at midstroke. When using the signal generator the setpoint signals are faded-in to guarantee a smooth transition.

Control loop

For each of the six hydraulic actuators a single control loop is available. The lay-out of the control loop comes down to the scheme as indicated in (Fig. 2.6).

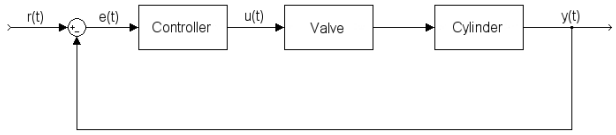


Figure 2.6: control loop

From a control point of view, the system behaves like a first-order system as it is controlled by flow, not by force. Therefore a simple K_p-controller is used to penalize the error between the reference piston position ($r(t)$) and the actual piston position ($y(t)$). The flow is controlled by means of adjusting the valve spool position, x_v, causing a change in flow out of the valve. For the valve itself a position feedback loop with a high bandwidth is present to control the valve spool position.

2.3 Measurements

In the simulation that will be presented in chapter 3 only one hydraulic actuator will be modeled. Therefore, the measurements are performed on a single actuator only. To be able to measure the pressure in the lower cylinder chamber a special block is inserted between the valve and the actuator on which a pressure transducer can be connected. The pressure in the upper cylinder chamber should be equal to the system pressure P_s, but it can also be measured directly by another pressure transducer on the valve manifold. To measure the acceleration of the single actuator a 3-DOF accelerometer is placed on the upper joint of the hydraulic cylinder that is connected to the motion platform. In the software there is a possibility to measure 2 extra signals, for example the actual piston position and the valve setpoint. These can be selected as the two test outputs and can be connected to the scope.

2.3.1 Smoothness

When the Micro Motion System is running in for example the heave-direction (=z-direction, see Fig. 2.1), the effect of the reversal bump caused by the overlapped valve and friction can be felt on the motion platform. This effect can be detected by looking at the acceleration profile, using the accelerometer. To be able to measure the acceleration of a single actuator, the following reference signal will be applied to one cylinder that starts from its neutral position, which is half the total stroke (equal to $\frac{L_s}{2}$):

$$y_{ref} = 0.05 \sin(\pi \cdot t) + \frac{L_s}{2} \tag{2.1}$$

The output of the accelerometer is a voltage that needs to be converted to the corresponding acceleration.

The accelerometer that is used has an offset of -3.3431 $[V]$ and a sensitivity of 0.826 $[\frac{V}{g}]$, so the acceleration can be calculated by:
$\ddot{y} = (U_{acc} - 3.3431) \cdot 0.826 \cdot 9.81$ $[\frac{m}{s^2}]$, with U_{acc} the output voltage of the accelerometer. The resulting actuator acceleration then looks like (Fig. 2.7), in which the deflection from the basic harmonic (magnitude of the peaks) is an indicative figure for the smoothness.

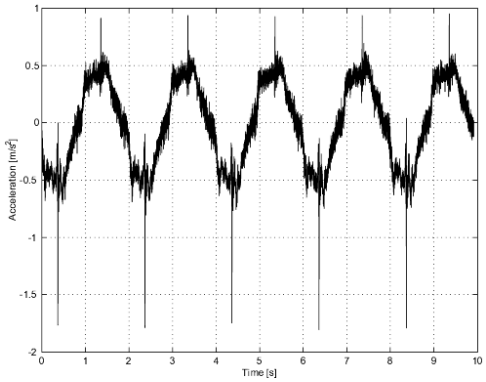


Figure 2.7: measured actuator acceleration

In this figure the effect of the reversal bump can be distinguished as peaks in the acceleration profile as a result of the application of an overlapped proportional valve.

2.3.2 Pressures

The pressures P_1 and P_s have also been measured during an experiment with the same setpoint, the results can be seen in (Fig. 2.8). In this figure it can be seen that the system pressure varies quite a

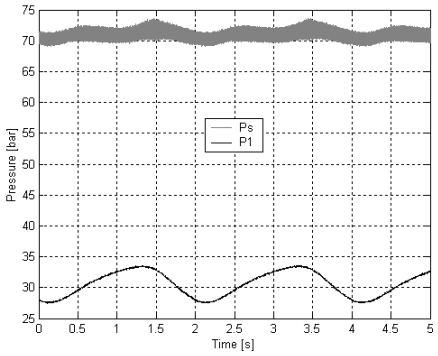


Figure 2.8: pressures

bit around the theoretical system pressure of 70 [bar]. The variation in the system pressure (which is the pressure in the upper cylinder chamber) depends on the reference signal the actuator has to track. Of course, when for example the cylinder moves down with a high speed the pressure in the upper cylinder chamber will temporarily decrease, because of the need for extra oil that cannot be there instantaneously. The pressure in the lower cylinder chamber (P_1) is in-phase with the acceleration profile.

2.3.3 Friction measurement

Another non-linearity in the hydraulic actuator is friction. To get an idea of the magnitude of the friction forces, experiments have been performed. The goal is to model the friction effects according to a practical Stribeck-model [6] that divides the friction forces into static friction, Coulomb friction and viscous friction. This Stribeck function is an empirical function:

$$F_f(\dot{y}) = \sigma\dot{y} + sign(\dot{y})[F_{c0} + F_{s0}\exp(-\frac{\dot{y}}{c_s})] \tag{2.2}$$

with:
$\sigma:\ viscous\ friction\ parameter\ [\frac{Ns}{m}]$
$F_{c0}:\ Coulomb\ friction\ parameter\ [N]$
$F_{s0}:\ static\ friction\ parameter\ [N]$
$c_s:\ Stribeck\ velocity\ [\frac{m}{s}]$

The parameters σ, F_{c0}, F_{s0} and c_s must be identified from experimental data. For this, the total actuator force is expressed as:

$$F_t = F_L(P_1, P_s) = M\ddot{y} + F_f + Mg\ \text{, so}:\quad F_f = F_L(P_1, P_s) - M\ddot{y} - Mg \tag{2.3}$$

Ideally, constant velocity experiments should be performed, because then the friction force can be determined by: $F_f = F_L(P_1, P_s) - Mg$. The stroke of the actuator is limited and therefore constant velocity experiments can only be carried out for small velocities. By measuring the friction force $F_f = F_L(P_1, P_s) - M\ddot{y} - Mg$ (with M the total mass of the piston rod and load of the motion platform) a model can be developed that covers a wider velocity-range. The pressure difference force is determined by $F_L(P_1, P_s) = P_1A_1 - P_sA_2$ with the pressures P_1 and P_s (system pressure) being the pressures in the lower and upper cylinder chamber respectively (see Fig. 3.1). These pressures are measured on the analog input channels of the scope and must be converted to their physical quantities, according to: $P_i = U_p\frac{315}{4}\cdot 10^5\ [\frac{N}{m^2}]$, with U_p the output voltage of the pressure transducer. By measuring the friction force at regular intervals as the cylinder is extended and retracted, it is then possible using the force equilibrium (Eq. (2.3)) to obtain the estimated friction force $\hat{F}_f(\dot{y})$. The results of the experiments can be given in a so-called friction-velocity map (see Fig. 2.9).

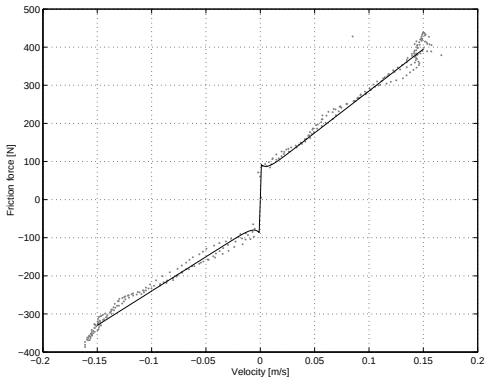


Figure 2.9: measured and approximated friction force

To capture the asymmetry of the friction forces (Eq. 2.2) can be extended to:

$$F_f(\dot{y}) = \begin{cases} \sigma^+\dot{y} + sign(\dot{y})[F^+_{c0} + F^+_{s0}\exp(-\frac{|\dot{y}|}{c^+_s})] & \forall\ \dot{y} \geq 0 \\ \sigma^-\dot{y} + sign(\dot{y})[F^-_{c0} + F^-_{s0}\exp(-\frac{|\dot{y}|}{c^-_s})] & \forall\ \dot{y} < 0 \end{cases} \tag{2.4}$$

From (Fig. 2.9) the parameters of (Eq. 2.4) can be obtained, the approximated friction curve is also indicated in the figure. The value of the parameters can be found in the numerical data sheet (see appendix A).

2.4 Discussion

All the different subsystems have been presented in this chapter as well as the software and control loop to drive the hydraulic system. With the aid of this setup measurements have been performed to indicate the smoothness of the Micro Motion System. Also the friction forces have been identified and parameterized according to an asymmetric Stribeck-function. This friction model will be used in the simulation model that will be developed in the next chapter.

Chapter 3

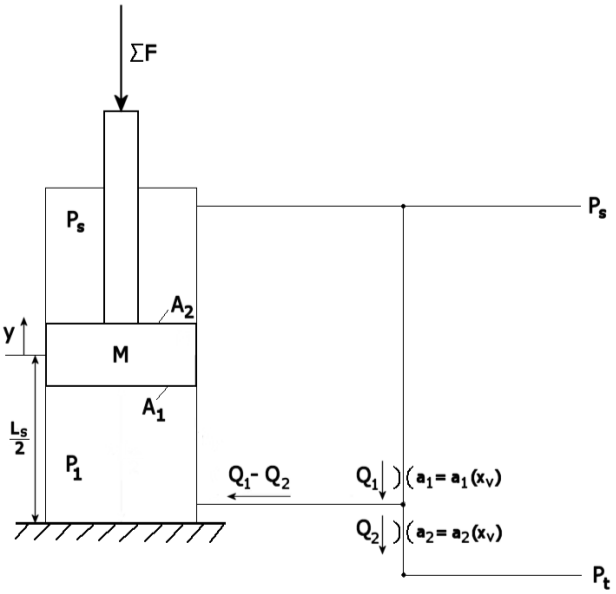
Modeling

3.1 Introduction

This chapter presents the modeling of the main parts that need to be taken into account to be able to investigate the smoothness of the hydraulic motion system. Therefore, assumptions (and simplifications) will be made to realize a set of equations that will describe the behavior of the actuator and valve. A Simulink model is created and the results of the simulation will be shown.

3.2 Assumptions and equations

To model the dynamics of the motion system the choice is made to use only one hydraulic actuator together with the proportional valve since the effect is the same for all actuators. Figure 3.1 shows the schematic representation of the hydraulic actuator and valve.



P_s: system pressure
P_1: pressure in lower chamber
a_i: flow-area
x_v: valve spool position
Q_i: flows in/out of the valve
A_1: piston surface area (bottom side)
A_2: piston surface area (upper side)
y: piston position
M: mass of piston
L_s: length of the stroke
$\sum F$: sum of forces acting on the piston

Figure 3.1: schematic representation

As can be seen, the pressure in the upper cylinder chamber is equal to the system pressure P_s. The different parts that need to be described by mathematical equations are: the valve dynamics, the flow out of the valve, the pressure dynamics (including continuity and compressibility of the oil) and the actuator dynamics. The pipeline dynamics between the valve and actuator can be neglected, according to Beater [1] the whole volume of the pipeline can be added to the corresponding chamber volume of the cylinder model if

$$l < \frac{c}{10 f_{max}} = \frac{340}{10 f_{max}} \tag{3.1}$$

with:
f_{max} : $the\ maximum\ interesting\ frequency\ [Hz]$
l : $length\ of\ the\ pipeline\ [m]$
c : $wave\ propagation\ speed\ (for\ oil)\ [\frac{m}{s}]$

Since $l = 65\ [mm]$, frequencies up to approximately 500 [Hz] can be investigated without having to consider the pipeline dynamics.

3.2.1 Valve dynamics

The position of the valve spool is controlled by a control-loop having a high bandwidth, so the valve can be modeled as a proportional gain relating the input voltage to a corresponding valve spool position:

$$x_v = u_v \frac{x_{v,max}}{u_{v,max}} \tag{3.2}$$

with:
x_v : $valve\ spool\ position\ [m]$
u_v : $input\ voltage\ [V]$
$u_{v,max}$: $maximum\ input\ voltage\ [V]$
$x_{v,max}$: $maximum\ valve\ spool\ position\ [m]$

The numerical values can be found in appendix A.

3.2.2 Valve geometry

As can be seen in (Fig. 3.1) the flow out of the valve depends on the valve spool position, x_v. Therefore a relation needs to be found to describe the flow area depending on the valve spool position. Figure 3.2 shows a schematic representation of the valve.

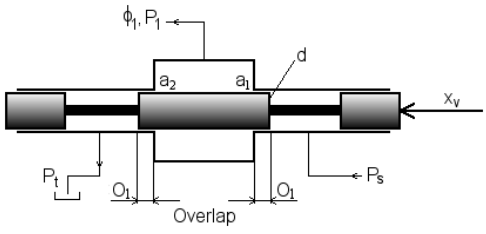


Figure 3.2: valve geometry (schematic)

The valve spool position x_v determines the valve openings $a_i(x_v)$ according to the following relations:

$$a_i(x_v) : \begin{cases} a_1(x_v) = \pi d(x_v - O_l) \ [m^2], & x_v > O_l \\ a_2(x_v) = -\pi d(x_v + O_l) \ [m^2], & x_v < -O_l \\ a_1(x_v) = a_2(x_v) = 0 \ [m^2], & -O_l \leq x_v \leq O_l \end{cases} \tag{3.3}$$

with:
$d: \ diameter \ of \ valve \ spool \ [m]$
$O_l: \ overlap \ \ at \ mid-position \ [m]$

3.2.3 Valve flow

For the oil-flow out of the valve the so-called orifice equation can be used. By assuming that the edges of the valve are sharp, causing a turbulent flow, the speed of oil through a narrow orifice is determined by $v_{oil} = C_d\sqrt{\frac{2\Delta P}{\rho}}$. The flow can now be found as the product of the port opening a_i (or flow-area) and the oil speed: $\phi_1 = a \cdot v_{oil} = a_i C_d\sqrt{\frac{2\Delta P}{\rho}}$. Writing down the relations for the three-way valve leads to:

$$\phi_1 = Q_1 - Q_2 = C_d a_1(x_v)\sqrt{2\frac{P_s - P_1}{\rho}} - C_d a_2(x_v)\sqrt{2\frac{P_1 - P_t}{\rho}} \tag{3.4}$$

with:
$C_d : discharge \ coefficient [-]$
$P_s : supply \ pressure \ [Pa]$
$P_t : tank \ pressure \ [Pa]$
$P_1 : pressure \ in \ bottom \ chamber \ [Pa]$
$\rho : density \ of \ oil \ (shell \ Tellus \ T46) \ [\frac{kg}{m^3}]$

3.2.4 Pressure dynamics

The pressure in the lower cylinder chamber can be calculated by looking at the mass conservation equation for a control volume V. Let the stored mass of the oil inside V be m with a density of ρ. Since the medium is assumed to be continuous, the rate at which the mass is stored must equal the incoming mass flow rate minus the outgoing mass flow rate:

$$\sum \dot{m}_{in} - \sum \dot{m}_{out} = \frac{d(\rho V)}{dt} = \rho \dot{V} + V\dot{\rho} \tag{3.5}$$

Suppose that variations in density and pressure are related through $\frac{\partial \rho}{\partial t} = \frac{\rho}{E}\frac{\partial P}{\partial t}$, based on the physical properties of the fluid involved. Substituting this relation into (Eq. 3.5), using $\dot{m} = \rho \cdot Q$ and dividing by ρ leads to:

$$\sum Q_{in} - \sum Q_{out} = \dot{V} + \frac{V}{E}\dot{P}, \text{ or } \dot{P} = \frac{E}{V}(\sum Q_{in} - \sum Q_{out} - \dot{V}) \tag{3.6}$$

Applying (Eq. 3.6) to the lower chamber of the hydraulic cylinder (see also Fig. 3.1), adding the volume of the pipeline according to (Eq. 3.1) and assuming no leakage flows, gives:

$$\dot{P}_1 = \frac{E}{yA_1 + V_{pl}}(\phi_1 - \dot{y}A_1) \tag{3.7}$$

with:
$E : bulkmodulus\ of\ oil\ [\frac{N}{m^2}]$
$y : piston\ position\ [m]$
$\dot{y} : piston\ velocity\ [\frac{m}{s}]$
$A_1 : piston\ surface\ area\ (bottom\ side)[m^2]$
$V_{pl} : volume\ of\ pipeline\ [m^3]$

3.2.5 Actuator dynamics

The most important forces acting on the actuator (see Fig. 3.1) are forces on the piston caused by the oil pressures P_1 and P_s, friction forces F_f and gravity forces acting on the total mass that needs to be displaced (mass of the piston and mass of the moving platform, actuator is in vertical position).

The following relation for the acceleration of the piston $\ddot{y}$ can then be determined by using Newton's law:

$$M\ddot{y} = P_1A_1 - P_sA_2 - F_f - Mg \tag{3.8}$$

To model the friction forces F_f the so-called Stribeck-curve [6] will be used to be able to identify Coulomb, static and viscous friction:

$$F_f(\dot{y}) = \begin{cases} \sigma^+\dot{y} + \arctan(k\cdot\dot{y})[F_{c0}^+ + F_{s0}^+\exp(-\frac{|\dot{y}|}{c_s^+})] & \forall\ \dot{y} \geq 0 \\ \sigma^-\dot{y} + \arctan(k\cdot\dot{y})[F_{c0}^- + F_{s0}^-\exp(-\frac{|\dot{y}|}{c_s^-})] & \forall\ \dot{y} < 0 \end{cases} \tag{3.9}$$

with:
$M : total\ mass\ of\ the\ piston\ rod\ and\ load\ [kg]$
$A_2 : piston\ area\ (topside)\ [m^2]$
$\ddot{y} : piston\ acceleration\ [\frac{m}{s^2}]$
$g : gravitation\ constant\ [\frac{m}{s^2}]$
$\sigma : viscous\ friction\ parameter\ [\frac{Ns}{m}]$
$F_c^+ : Coulomb\ fricton\ parameter\ for\ positive\ velocities\ [N]$
$F_c^- : Coulomb\ friction\ parameter\ for\ negative\ velocities\ [N]$
$\sigma^+ : viscous\ friction\ parameter\ for\ positive\ velocities\ [\frac{Ns}{m}]$
$\sigma^- : viscous\ friction\ parameter\ for\ negative\ velocities\ [\frac{Ns}{m}]$
$F_{s0}^+ : static\ friciton\ parameter\ for\ positive\ velocities\ [N]$
$F_{s0}^- : static\ friction\ parameter\ for\ negative\ velocities\ [N]$
$c_s^+ : Stibeck\ velocity\ parameter\ for\ positive\ velocities\ [\frac{m}{s}]$
$c_s^- : Stribeck\ velocity\ parameter\ for\ negative\ velocities\ [\frac{m}{s}]$

The parameters F_c^+, F_c^-, σ^+, σ^-, F_{s0}^+, F_{s0}^-, c_s^+ and c_s^- have been identified in section 2.3.3. In (Eq. 3.9) the $arctan$-function is preferred instead of the $sign$-function, because the use of the highly non-linear $sign$-function could lead to numerical problems in the simulation (the $arctan$-function is smooth around $\dot{y} = 0$). Of course, this depends on the solver that will be used in the simulation, which will be discussed in the next section.

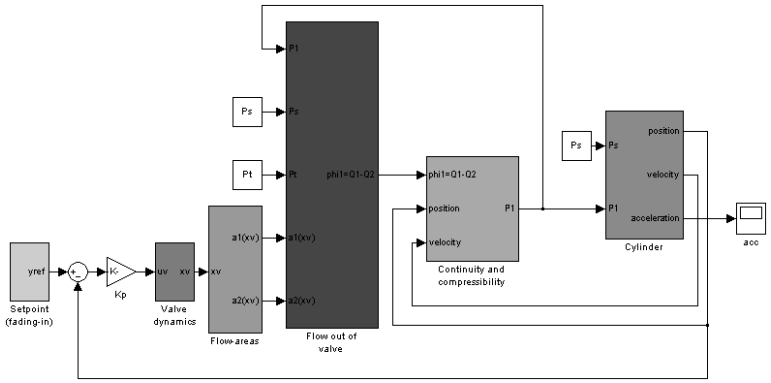


Figure 3.3: Simulink model

3.3 Simulink model

The equations that have been described until now can be put into a Simulink model (see Fig. 3.3).

The different subsystems can be seen in appendix A. To describe the relations for the flow-areas according to (Eq. 3.3), an s-function is used. The choice of reference signal plays an important role in the simulation. To circumvent the problem that an initial velocity unequal to zero occurs caused by the choice of for example a sine wave, this sine wave will be faded in just like the real system. The reference signal that will be used is:

$$y_{ref} = 0.05 \sin(\pi \cdot t) + \frac{L_s}{2} \tag{3.10}$$

with $\frac{L_s}{2}$ indicating the start from neutral position. This reference function will be multiplied by a ramp signal that will be equal to one after one period of this sine wave, so the fade-in time will be 2 [s]. The advantage of this is that $\dot{y}_{ref}|_{t=0} = 0[m/s]$, to prevent the actuator having to react on an instantaneous velocity, which is not physically achievable. This leads to another important issue, namely the initial conditions for the three integrators that are present in the Simulink model. Initial conditions are required for:

1. $y|_{t=0} = \frac{L_s}{2}\ [m]$. The actuator starts from its neutral position.
2. $\dot{y}|_{t=0} = 0\ [\frac{m}{s}]$. Due to the fading in of the setpoint the initial velocity will be zero.
3. $P_1|_{t=0} = (P_s - P_t)\frac{A_2}{A_1} + \frac{Mg}{A_1}$. This condition can be derived by substituting $\ddot{y} = 0\ [\frac{m}{s^2}]$ and $\dot{y}|_{t=0} = 0\ [\frac{m}{s}]$ into (Eq. 3.8), because the forces acting on the piston have to be in equilibrium at $t = 0$.

By using the parameters out of the parameter list (see appendix A), a simulation can then be performed. The simulation can be run either by using a solver with a variable step time (and by setting the right tolerances) or using a fixed step time. After some trials a small fixed step time has been chosen to make sure the equations can be solved, therefore the simulation runs with a sampling frequency of 10 kHz. Some results of the simulation are shown in (Fig. 3.4). Fading-in of the setpoint can be seen in the upper left-hand picture in this figure, the actuator moves around its neutral position ($\frac{L_s}{2}$). The feedback gain, K_p (see Fig. 3.3) is tuned by hand, resulting in a tracking error as indicated in the top right-hand picture. This could be improved, but this is not the main issue of the simulation. The lower left-hand picture in (Fig. 3.4) shows the flow out of the valve. The piston velocity is proportional to this flow. Finally, the pressure in the lower cylinder chamber is shown, it looks like there is a transient in this signal, the cause of this will be shown later on. This pressure P_1

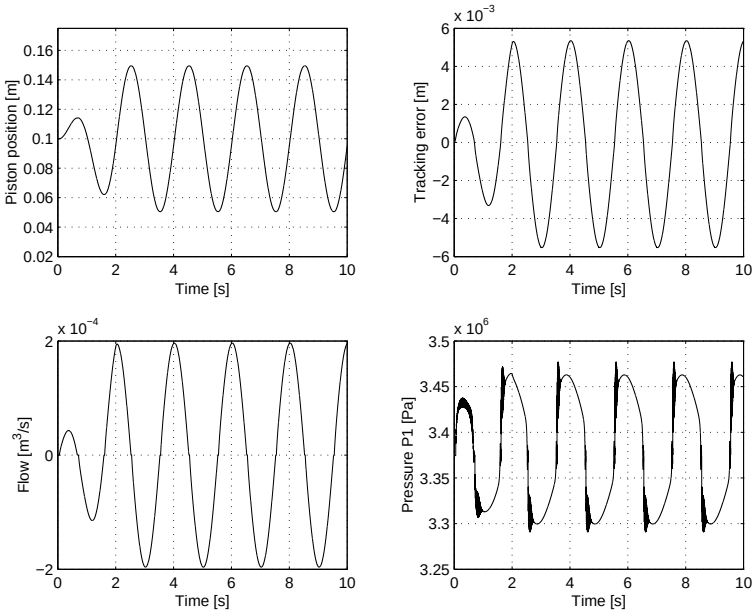


Figure 3.4: simulation results

is closely related to the acceleration of the piston as can be seen on the left-hand side of (Fig. 3.5). This figure shows the resulting acceleration profile caused by a single hydraulic actuator. Again, a transient seems to be present, as can be seen in the left-hand plot of (Fig. 3.6) in more detail. To investigate the cause of this a Fourier Transformation of the acceleration profile is given in the right-hand plot of (Fig. 3.6). Here it can be seen that a peak occurs at the eigenfrequency of the actuator, which lies at $f_e = \frac{1}{2\pi}\sqrt{\frac{2EA}{ML_s}} \approx 109\ [Hz]$ as discussed in [8]. By looking at the measured actuator acceleration profile (see Fig. 2.7) this transient cannot be distinguished. To reduce the transient that occurs in the simulation either the system parameters should be adapted or the flow-curve needs to be investigated. In this case, the flow-curve will be investigated since the parameters are chosen realistically. The right-hand plot of (Fig. 3.5) shows the flow-curve, indicating the dead-band behavior around $u_v = 0\ [V]$. The dead-band behavior causes the peaks in the acceleration profile. As the acceleration is the first derivative of the velocity and this velocity is proportional to the flow out of the valve, which in turn depends on the flow-area function that needs to be investigated.

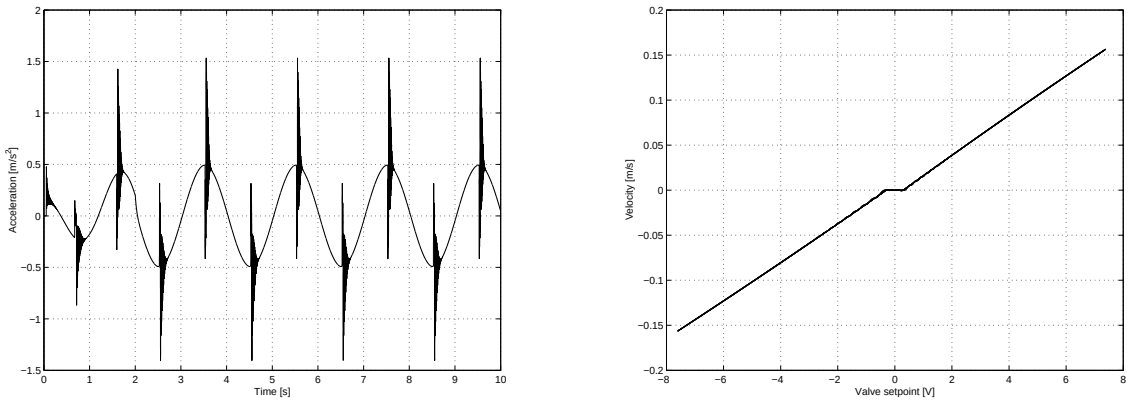


Figure 3.5: acceleration profile (left) and flow-curve (right)

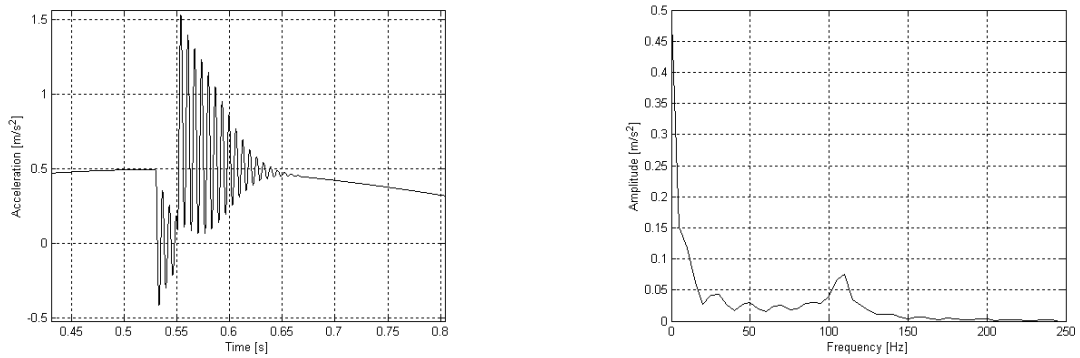


Figure 3.6: transient (left) and fft (right)

This function, according to (Eq. 3.3), is shown in (Fig. 3.7).

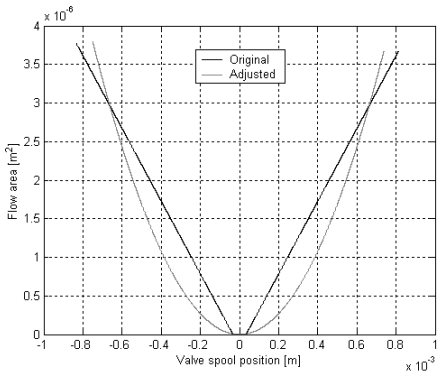


Figure 3.7: flow-area function

As can be seen in the figure, the flow-area function is not very smooth when the valve is around its mid-position (the hard bound of $x_v \geq O_l$ and $x_v \leq -O_l$). A function $f(x)$ is *smooth* if it is continuously differentiable up to any order of x (see [3]), which is not the case here. This might be the cause of the transient in the acceleration profile, because this sudden change causes a step in the flow, resulting in a swing-up of P_1 due to the stiffness of the differential equation (Eq. 3.7). By using a smooth flow-area function this swing-up might be reduced. Figure 3.7 also shows another possible flow-area function, which is smooth (also around $x_v = \pm O_l$). The new equation that will describe the flow-area is:

$$a_i(x_v) = 300 \cdot x_v^2 \tag{3.11}$$

The gain of 300 is required to make this function correspond with the previously used flow-area function. Now, the results of a simulation with the flow-area function according to (Eq. 3.11) can be evaluated.

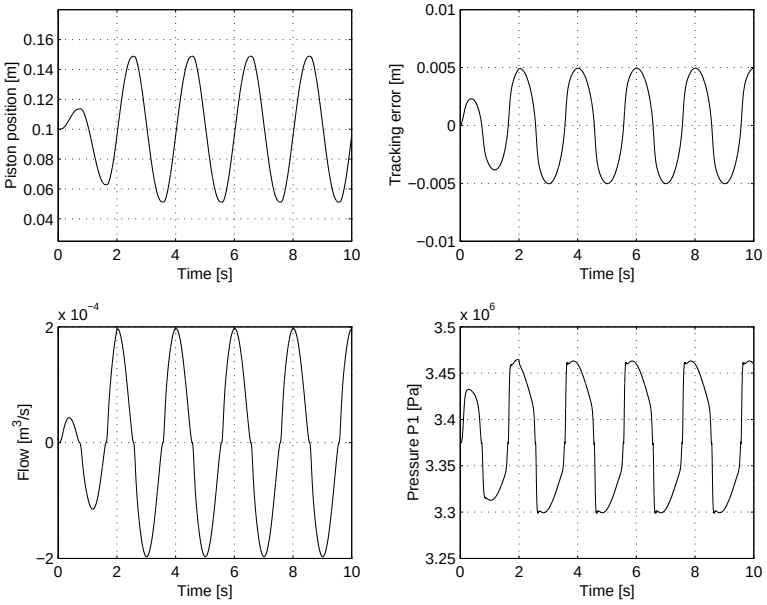


Figure 3.8: simulation results

Figure 3.8 again shows the simulation results of the actual piston position y, the tracking error e, the flow ϕ_1 out of the valve and the pressure in the lower cylinder chamber, P_1. It can be seen that now the transient in $P_1(t)$ has disappeared.

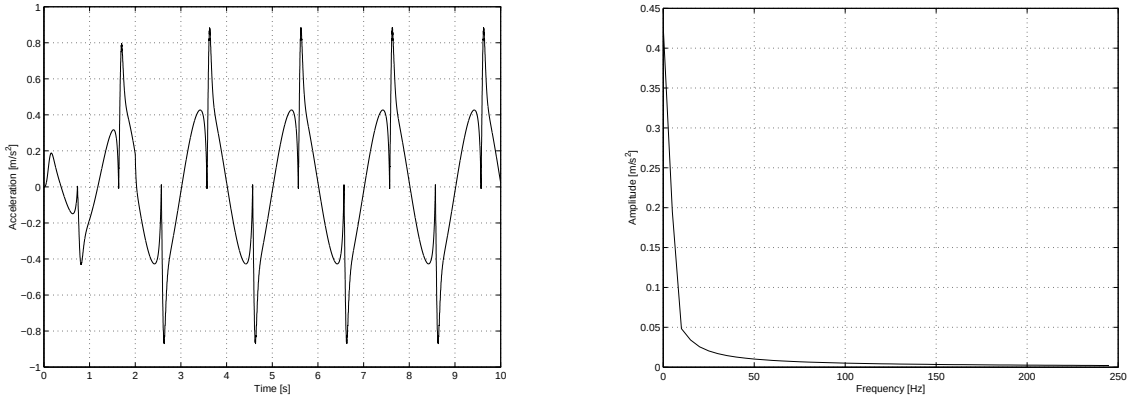


Figure 3.9: acceleration profile (left) and fft (right)

The acceleration profile is indicated in the left picture of (Fig. 3.9), showing a more realistic behavior as now two peaks can be recognized each time the valve spool passes $x_v = 0$. This corresponds with the change of sign of the derivative of the flow-curve (this can also be seen in the velocity profile), showing the true effect of the so-called 'reversal bump'. In the right-hand picture of (Fig. 3.9) again the Fourier transformation of the acceleration profile is plotted, showing that the eigenfrequency is not excited with this adjusted flow-area function. The resulting flow-curve can be seen in (Fig. 3.10).

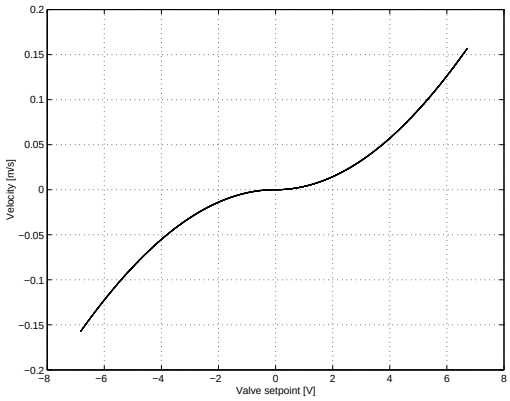


Figure 3.10: flow-curve

3.4 Discussion

This chapter included the modeling of the different parts of the hydraulic actuator. Some assumptions and simplifications have been made to come to a set of equations that describe the behavior of the actuator. Together with the numerical data simulations have been performed. The choice of flow-area function is of great influence on the quality of the simulation. Hard non-linearities, not only for the flow-area function bur also for example a sign-function, should be avoided to guarantee realistic simulation results and prevent numerical problems. These problems have been shown in this chapter and also a solution to circumvent these has been found by the choice of smooth functions. The goal of the simulation is reached, namely to visualize the effects of the overlap in the proportional valve. The effect of the reversal bump on the smoothness has been presented and this simulation will be compared to the experimental results in the next chapter.

Chapter 4

Simulation and reality: the differences

4.1 Introduction

Although the goal of the simulation is to visualize the main actuator dynamics and not to exactly represent the behavior of the complete system, it is interesting to acknowledge the main differences between the simulation (with its assumptions and simplifications) and reality. Before a control algorithm will be developed, results of the experiments on the real system and the simulation results will be compared. Main aspects are of course the smoothness of the system, the flow curve and other characteristic signals such as the pressures in both cylinder chambers.

4.2 Smoothness

First of all, the measured acceleration profile will be compared to the simulated one (see Fig. 4.1). As

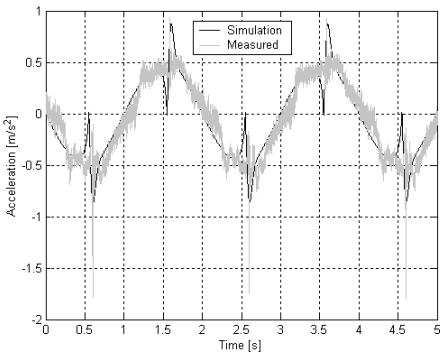


Figure 4.1: acceleration profile in simulation and reality

can be seen, the peaks in the acceleration profile occur at approximately the same time. Of course, the measured acceleration is subject to measurement noise. It can also be seen that the magnitude of the peaks in the measured acceleration profile are larger at the minima of this profile. This can be declared by the fact that at that time the actuator velocity changes sign from positive to negative, so this

occurs when the actuator position is at its maximum. Gravity plays an important role here, because when the velocity changes sign the gravity suddenly 'helps' the actuator going down, causing a slight extra step in the acceleration. A reason for the difference between the two acceleration profiles lies in the fact that in the simulation the actuator is assumed to be in vertical position. In reality a single actuator cannot be driven into a vertical position, because of the limited stroke of the actuator when it is connected to the motion platform.

4.3 Pressures

One of the assumptions that have been made is that the pressure in the upper cylinder chamber is equal to the system pressure P_s, which is assumed to be constant. As could be seen in (Fig. 2.8) this is not the case in reality. The system pressure varies sinusoidally according to the reference signal that has to be followed and this effects P_1, see (Fig. 4.2). According to (Eq. 3.7), P_1 depends on the

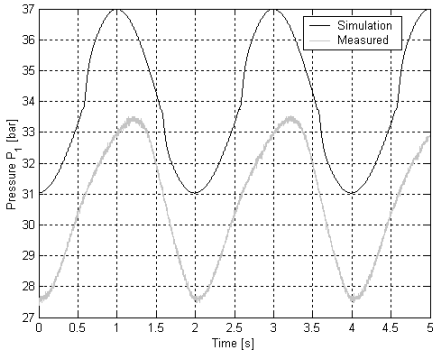


Figure 4.2: pressure P_1, simulation and reality

piston position and piston velocity, which in turn are calculated out of the acceleration (Eq. 3.8) in the simulation. This acceleration depends on the difference between P_1 and P_s, so because of the fact that P_s is constant, the effect of the friction forces can be seen in the evolution of $P_1(t)$. The reason for the difference in magnitude of the measured and simulated lower cylinder chamber pressure lies in the slight deviation of the magnitude of the system pressure P_s and the approximated total mass of the piston and load. This load consists of that part of the moving platform that rests on a single actuator, together with the joint that connects them. So, these deviations can easily be corrected in the simulation parameters, but the magnitude of all the main effects is not of great importance, because the simulation is made to *visualize* the main aspects and effects of a hydraulic actuator controlled by a proportional valve.

4.4 Flow-curve

To measure the piston position of the hydraulic actuator a position transducer is used. To be able to measure the piston velocity this position signal must be differentiated with respect to time. A good velocity measurement is quite difficult, because of the measurement noise on the position signal. A way to obtain the velocity is to first filter the position signal with a low-pass filter using the 'filtfilt'-function in Matlab before differentiating. By doing so it is possible to perform zero-phase digital filtering by processing the input data in both the forward and reverse directions. After filtering in the forward direction, the filtered sequence is reversed and runned back through the filter. The resulting sequence has precisely zero-phase distortion and double the filter order. Of course this filter method

can only be done off-line, since the 'filtfilt'-function uses information in the signal at points before and after the current point, in essence 'looking into the future' to eliminate phase distortion, so this is an anti-causal implementation. A drawback of this method is that some relevant information (higher frequencies) will be lost. As a consequence, the dead-band behavior might not be properly reproduced. The valve setpoint is filtered with the same filter to obtain a flow-curve as realistic as possible, resulting in the flow-curve as indicated in (Fig. 4.3). Here, the dead-band behavior cannot be distinguished, but

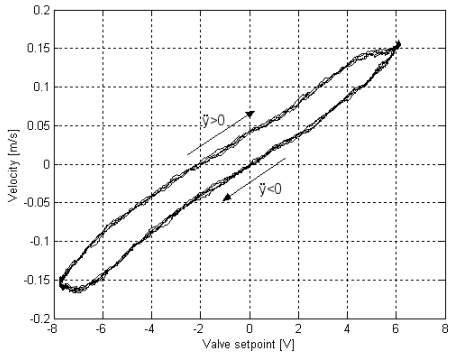


Figure 4.3: flow-curve

it certainly exists according to the measured acceleration profile! This is the result of filtering with a low-pass filter (high-frequency information will be lost), but it is the only way to achieve the velocity signal. Also, the effect of hysteresis can be seen in (Fig. 4.3) caused by internal friction, which depends on the direction of motion of the actuator (upwards or downwards). This implies that the behavior of the system is different for $\ddot{y} > 0$ and $\ddot{y} < 0$ with this setpoint.

4.5 Discussion

This section showed the main differences between the simulation and reality. The smoothness of the system can be described by looking at the acceleration profile in the simulation model, although some slight differences still remain. This is also caused by the measurement devices itself, as is the problem in determining the flow-curve of the actuator. A good velocity measurement is difficult, this is why the real dead-band behavior cannot be recognized in the results. So, although there are some differences, especially in the magnitude of the main system signals, the simulation model can be used to describe the main non-linearities (reversal bump, friction) and act as a starting point to determine a compensation algorithm.

Chapter 5

Compensation algorithm: the possibilities

5.1 Introduction

In chapter 3 a Simulink model has been derived, indicating the smoothness problem. The goal is to improve the smoothness of a single actuator by using an adaptive feedforward algorithm. Therefore, the basic principle will be explained before a compensation algorithm will be presented. Of course, this algorithm needs to be tested in the simulation, leading to new insights and room for improvement. So, modifications will be made and conclusions will be drawn from the different simulation results.

5.2 Compensation in theory

Due to the overlap in the valve there exists a non-linearity between the valve setpoint and the velocity of the piston rod. To circumvent this problem the valve compensation due to Polzer and Nissing [4] can be used. For this purpose, the piston velocity for an arbitrary input voltage is measured for the static case, characterized by $\ddot{y} = 0$ and $\dot{P}_1 = \dot{P}_s = 0$. By means of the measured pressures it is possible to calculate a theoretical piston velocity that depends on the valve setpoint u_v. The required relationship can be derived from combining (Eq.3.2), (Eq. 3.4), (Eq. 3.7) and (Eq. 3.11):

$$\dot{y}_{des} = \begin{cases} \frac{300 C_d x_{v,max}^2}{A_1 u_{v,max}^2} \sqrt{2 \frac{P_s - P_1}{\rho}} u_v^2 & \forall\ u_v > 0 \\ \frac{300 C_d x_{v,max}^2}{A_1 u_{v,max}^2} \sqrt{2 \frac{P_1 - P_t}{\rho}} u_v^2 & \forall\ u_v < 0 \end{cases} \tag{5.1}$$

which depends on the direction of motion, since the piston's surface area is different for positive and negative velocities.

A typical $(u_v, \dot{y})$-curve is given by the dotted line in (Fig. 5.1). Also in (Fig. 5.1) the simulated flow-

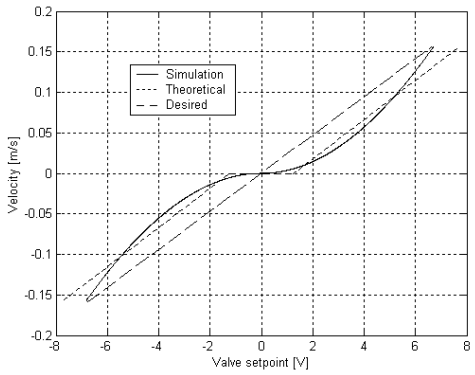


Figure 5.1: simulated, typical theoretical and desired flow-curves

curve is shown (dashed line), indicating the difference between a typical flow-curve and the one used in the simulation according to the changes in flow-area functions, using (Eq. 3.11) instead of (Eq. 3.3) for improving the simulation results as explained in section 3.3. Now, the goal is to force the non-linear relationship between the valve setpoint u_v and the piston velocity $\dot{y}$ into a linear one. This can be seen by looking at the *simulated* flow-curve, indicated by the solid line in (Fig. 5.2). An arbitrary input

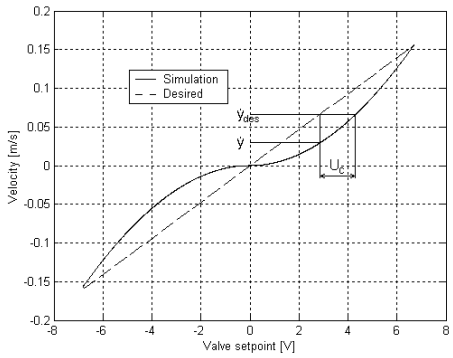


Figure 5.2: initial and desired flow-curves

voltage leads to a velocity $\dot{y}$ whereas ideally $\dot{y}_{des}$ should be reached. This desired velocity corresponds to a larger valve setpoint according to the simulated flow-curve. So, for every u_v a compensation-voltage U_c is required, which depends on the difference between the actual piston-velocity $\dot{y}$ and the desired piston velocity $\dot{y}_{des} = s \cdot u_v$, with s the slope of the desired curve.

The compensation term U_c can be added to the control effort to compensate for the dead-band behavior (see Fig. 5.3). Now, the required compensation voltage can be computed as the difference be-

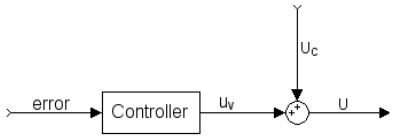


Figure 5.3: valve correction

tween the uncompensated flow-curve and the desired or zero-lap curve. From (Fig. 5.1) the difference in shape of the correction curves for the typical theoretical flow-curve and the simulated flow-curve can be recognized as in (Fig. 5.4). The required shape of the compensation curve when using this par-

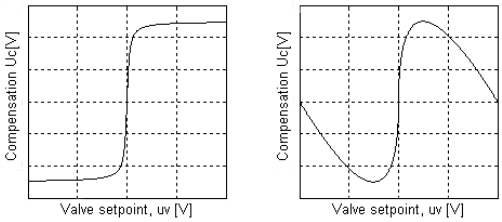


Figure 5.4: compensation for theoretical (left) and simulated (right) flow-curve

ticular flow-area function can be computed by looking at (Eq. 5.1). This relation is equal to $C \cdot u_v^2$, with $C = \frac{300 C_d x_{v,max}^2}{A_1 u_{v,max}^2} \sqrt{2 \frac{P_s - P_1}{\rho}}$ for $u_v > 0$ and $C = \frac{300 C_d x_{v,max}^2}{A_1 u_{v,max}^2} \sqrt{2 \frac{P_1 - P_t}{\rho}}$ for $u_v < 0$, with C a constant if $\ddot{y} = \dot{P}_1 = 0$. To make the relation between u_v and $\dot{y}$ linear, the compensation voltage u_c must be equal to $\sqrt{u_v} - u_v$, leading to $C \cdot u_v$, in which C is equal to the slope of the desired flow-curve. Of course, this required compensation voltage can be computed and implemented as a look-up table, but it would be nice if this could be done automatically whilst the system is moving. Another important aspect is that the required compensation varies during system operation due to for example temperature changes, so it also needs to be adapted when required. This will be treated in the next section.

5.3 Compensation in simulation

There is a demand for an algorithm that adjusts and stores the required compensation voltage U_c for every valve setpoint u_v. By doing so it is possible to update the compensation for every point on the flow-curve until the desired relationship is reached. Depending on the range of the valve setpoint (for example $\pm 10\ [V]$), a choice can be made for the buffer length to store the compensation. In this case, the bufferlength l (or vectorlength) is chosen to be equal to 1001 elements. This implies that the first 500 elements can be used to store the compensation for $u_v < 0$, element number 501 for $u_v = 0$ and elements 502 to 1001 for $u_v > 0$. This vector forms the basis for the c-file s-function in which the compensation voltage will be computed (in a Matlab environment). The slope s of the desired relationship between u_v and $\dot{y}$ could be computed in advance out of a measurement of the original flow-curve or it can be computed while the system is moving. An estimate for the slope could be: $s = \frac{max(\dot{y}) - min(\dot{y})}{max(u_v) - min(u_v)}\ for\ t \to \infty$.
The first condition that has to be fulfilled before the compensation vector must be updated is that there is a difference between the actual and desired flow-curves. Then, the element out of the vector that needs to be updated can be determined. It is possible that for different successive valve setpoints

the same element will be computed, corresponding with two slightly different compensation voltages, which is a drawback of this method. To create a compensation that won't react too nervous, the buffer needs to be updated once for every element. This can be achieved by comparing the actual vector-element with the previous vector-element, which is the actual vector element delayed by T_s (sample time). To create memory in the c-file 2 buffers will be used, namely $Bufferold$ and $Bufernew$, both of length l. Then, the update can be computed by adding the actual difference between the desired and actual velocity (multiplied by a penalizing gain) to the previously stored compensation (in the old buffer). By increasing the gain a faster convergence could be achieved, but care has to be taken of multiplication of the noise that may be present. The computed compensation is an output of the s-function and this value is then set to be equal to the old compensation. If the same vector element is computed successively the old compensation is sent to the output of the s-function (no updating). A drawback of this algorithm is that the compensation that will be computed can be rather noisy, because it depends on u_v and $\dot{y}$ that can both contain a certain amount of measurement noise. Another drawback is the use of the rounding-function, causing small steps in the computed compensation out of the s-function (see Fig. 5.5). A good way to solve this is to filter the computed compensation, but

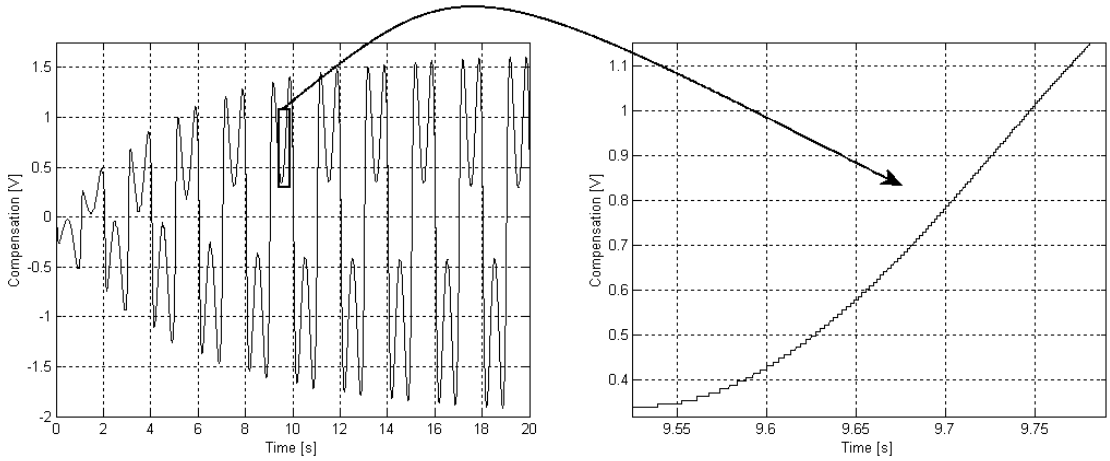


Figure 5.5: non-smooth computed compensation

because of the phase-loss, this compensation cannot be used directly as offset to the valve-setpoint. Also an interpolation could be performed, but the problem of measurement noise still remains. There is a different option, because of the fact that the *shape* of the compensation function is known, which is indicated in (Fig. 5.4). This offers the possibility to use the shape of the computed compensation curve to find the parameters of the algebraic compensation function. By using this parameter-adjustment filtering of the computed compensation is possible, because the phase is not important now. This procedure will be tested in the simulation. In order to use the shape of the computed compensation curve to find the parameters that describe the compensation function, two possible algebraic functions will be compared:

$$(1): \; u_{c1} = a\sqrt{\sin(b_1 u_v - c)} \tag{5.2}$$

and:

$$(2): \; u_{c2} = 4a[\sqrt{b_2 u_v - c} - (b_2 u_v - c)] \tag{5.3}$$

Of course, when implementing the compensation functions (1) and (2) described by (Eq. 5.2) and (Eq. 5.3), care has to be taken of the square-root for negative input values (this is done by taking the square-root out of the absolute value and multiplying by the sign of the original expression). Both functions are normalized (this is the reason for the factor 4 in (Eq. 5.3)), which helps to be able to use the computed parameters directly in these functions. The shape of these compensation functions (depending on the different parameters) can be seen in (Fig. 5.6). The first parameter, a, corresponds to the global maximum of U_c. To find a it is necessary to compare the actual compensation voltage U_c

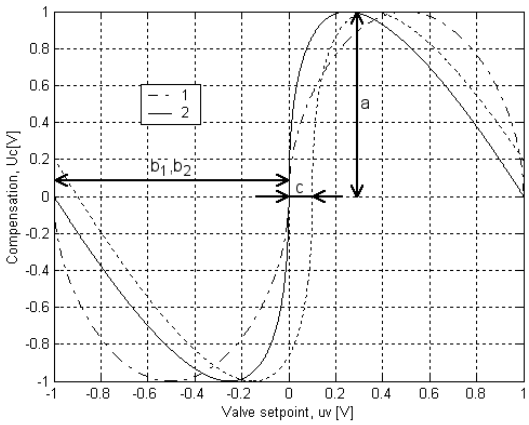


Figure 5.6: algebraic compensation functions

with the previous compensation voltage U_{cd} (=U_c delayed by T_s). So, by doing this and by sending U_c to the output of the s-function as long as $U_c > U_{cd}$ and keeping the maximum value of the difference when $U_c \leq U_{cd}$, a can be identified. Comparing, holding and output of the difference is done by a special Simulink-block (see Fig. B.4 in appendix B).

The computation of parameter b_1 in function (1) (Eq. 5.2) is related to the maximum valve input voltage, which indicates the value of u_v (indicated by q) when $u_{c1} = \sqrt{\sin(b_1 u_v - c)} = 0$ apart from $u_v = 0$. This is true if $b_1 u_v = k \cdot \pi$ (assuming c to be small initially) and only the value for $k = 1$ is required, so $q = \frac{\pi}{b_1}$ and an approximation of this value is given by $q = max(\frac{u_{v,max} - u_{v,min}}{2})$. Then, b_1 can be determined by $b_1 = \frac{\pi}{q}$, with $q \neq 0$. The computation of parameter b_2 in function (2) (Eq. 5.2) is also related to the maximum valve input voltage, but now b_2 indicates the value of u_v when $u_{c2} = 4a[\sqrt{b_2 u_v - c} - (b_2 u_v - c)] = 0$ (again indicated by q). Apart from $u_v = 0$ this is true if $\sqrt{b_2 q} = b_2 q$ (again assuming c to be small or zero initially). So, an approximation of b_2 is given by $b_2 = \frac{1}{q}$, with $q \neq 0$. At start of the system, q will be zero and therefore b_1 and b_2 cannot be determined. So, a threshold value for b_1 and b_2 has to be used to solve this problem (kind of an initial guess). After some time q reaches this value and then its actual value can be used to determine the parameters of the compensation function. The horizontal displacement of the compensation curve is guarded by parameter c. The displacement can be caused by heating in the valve and thereby causing a change of friction force acting on the valve spool or a change in the viscosity of the oil, which implies that a larger or smaller input voltage is required to reach a specific valve spool position than before (for example at start of the system ($t = 0\ [s]$)). Parameter c is related to the voltage u_v when $\dot{y} = 0$, guarded by the parameter r, this is where the steepest slope of the compensation is required, namely at the time of the reversal (change of the sign of $\dot{y}$). So, $r = u_v|_{\dot{y}=0}$ is the theoretical value , but it cannot be guaranteed that the numerical value of $\dot{y}$ will be equal to zero at one time. Therefore it is better to look at a small interval around $\dot{y} = 0$, so $k^- \leq \dot{y} \leq k^+$, with k^- and k^+ the lower and upper bounds of the region around zero, respectively (for example $k^- = k^+ = 10^{-3}\ [\frac{m}{s}]$). The parameter r can be computed by: $r = \frac{u_{v,max}|_{k^- \leq \dot{y} \leq k^+} - u_{v,min}|_{k^- \leq \dot{y} \leq k^+}}{2}$. Then, it should be guaranteed that the compensation function is equal to zero at the voltage u_v when $\dot{y} = 0$, taking into account the value of b_1 or b_2, so the parameter c will be: $c = b_1 r$ or $c = b_2 r$ (depending on the compensation function that is used).

5.4 Implementation

The implementation of the compensation algorithm (as discussed in the previous section) in the original Simulink model can be seen in the appendix in (Fig. B.5). In this model all the different parts can

be recognized, as discussed in chapter 3. Also two s-functions can be recognized, called 'function1' (this file contains the c-code with the two buffers to iteratively compute the required compensation, as discussed before) and 'function2' (this file contains the c-file to compute the smooth compensation function (1) or (2), according to (Eq. 5.2) and (Eq. 5.3). The same reference signal as described by (Eq. 3.10) will be used, which again will be faded in to make sure the same initial conditions hold.

5.4.1 Results of simulation (1)

To compare both compensation functions simulations will be performed. First of all, the results of the simulation with the compensation function described by (Eq. 5.2) will be shown, these can be seen in (Fig. 5.7) and (Fig. 5.8). In (Fig. 5.7) the flow-curve, parameter convergence (with parameter q being

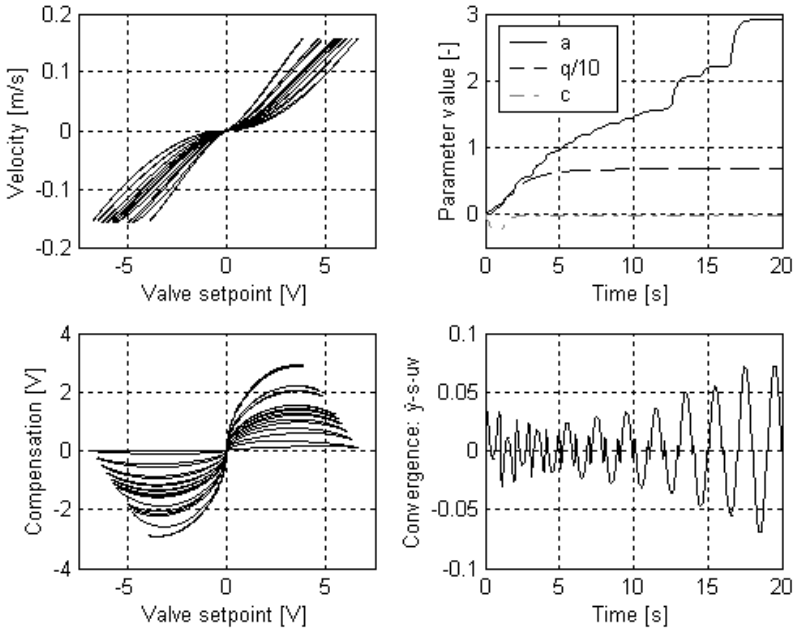


Figure 5.7: simulation results with compensation function (1)

scaled to the magnitude of the other parameters), compensation curve and convergence of $\dot{y} - s \cdot u_v$ is shown. By looking at the flow-curve the desired linear relationship is not reached, this indicates that $\dot{y} - s \cdot u_v$ does not converge. So, this implies that a constant error remains between $\dot{y}$ and u_v. Because of the update rule that is used in the c-file (called 'function') the computed compensation will blow up because of this error. As a result of this, parameter a will increase; causing a compensation that will become too large.

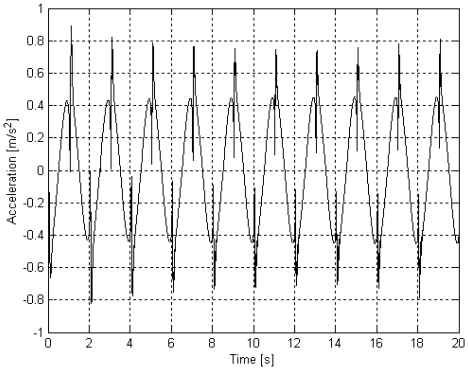


Figure 5.8: acceleration profile

In (Fig. 5.8), it can be seen that initially the peaks in the acceleration profile decrease, but these will increase because of the ever-increasing compensation due to parameter a. To prevent the compensation being too large, parameter a could be limited, but this implies that the resulting smoothness satisfies the demands (not optimal). Before any conclusions are drawn, the results of the simulation using the second compensation function will be presented.

5.4.2 Results of simulation (2)

The results of the simulation with the compensation function described by (Eq. 5.3) can be seen in (Fig. 5.9) and (Fig. 5.10).

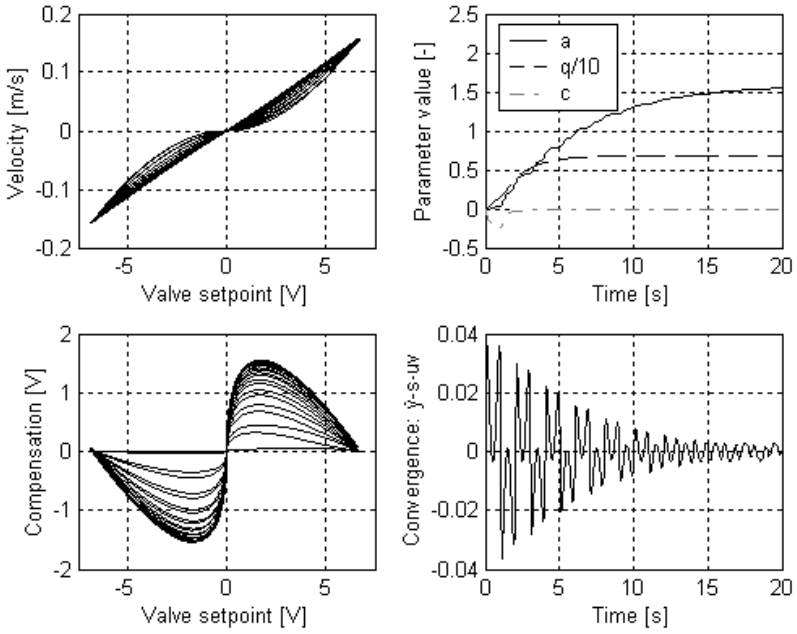


Figure 5.9: simulation results with compensation function (2)

Figure 5.9 again shows the flow-curve, parameter convergence, compensation curve and convergence of $\dot{y} - s \cdot u_v$. Now it seems that the desired linear relation between $\dot{y}$ and $s \cdot u_v$ can be reached. So, it looks like the second algebraic compensation function is better suited to compensate the effects of the quadratic flow-area function that is used in this simulation, as can be detected by looking at(Eq. 5.1). By looking at the differences between the compensation functions (1) and (2) as indicated in (Fig. 5.6), the main difference lies in the value of the compensation voltage around $u_v = 0$, before the compensation curve starts to bend. So, the voltage that is required for small deviations from $u_v = 0$ cannot be reached with compensation function (2). In other words, the slope of the curve around $u_v = 0$ is not enough to compensate for the occurring dead-band behavior. With compensation function (2) the required compensation voltage can be reached, so the difference between $\dot{y}$ and $s \cdot u_v$ decreases as the parameters evolve. This results in a significant decrease of the peaks in the acceleration profile (see Fig. 5.10); the smoothness is increased considerably.

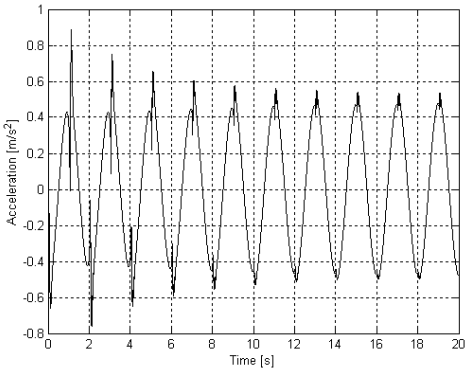


Figure 5.10: acceleration profile

5.4.3 Discussion

An important issue is the choice of the threshold value for the parameters b_1 and b_2. These parameters indicate the range of input-voltages that should be covered for computing a compensation voltage. This threshold value should not be chosen too low, because it results in a small period of the (u_v, U_c)-curve and because of the slowly evolving parameters a wider compensation curve could be necessary whilst larger valve setpoints are already passed to the system. On the other hand, by looking at the setpoints that will be applied to the system the valve input range can be estimated beforehand. As can be seen in the simulation results, the choice of the algebraic compensation function plays an important role. With compensation function (1) the desired linear behavior cannot be accomplished (upper left plot in Fig. 5.7), so the use of the iterative process together with this function is probably not the right choice. When convergence cannot be guaranteed, care has to be taken of the computed compensation to prevent similar problems as indicated in the first simulation. This asks for a safer algorithm that will lead to the maximum smoothness that can be achieved by a specific algebraic function, which will be discussed in the next section.

5.5 A different approach

Because of the fact that it cannot be guaranteed that a compensation function can be found that cancels the non-linearity caused by the overlap in the valve, a different approach is required that leads to a safer algorithm to prevent problems that might occur when the parameters fail to converge (see section 5.4.2). This approach is based on computing parameters over a certain amount of time, which means that by storing the relevant data ($u_v,\ \dot{y}$) in buffers once in a user definable time the parameters can be computed and sent to the compensation function. This also has the advantage of being able to cancel any outliers (filtering) that may be present in the different signals and to be able to use weighting functions. The algorithm will be implemented in an m-file s-function (to be able to use the standard Matlab-functions, see appendix C for the sourcecode). A total of three buffers is required, namely UV to store the valve setpoints, UC to store the computed compensation out of the actual difference between u_v and $\frac{\dot{y}}{s}$ (with s the desired slope of the flow-curve) and UY to store the valve setpoint when the velocity is in an interval around zero as described before. Initially all these buffers are empty and until a bufferlength l (corresponding to a certain time-range) is reached these buffers will be filled. Once the pre-defined bufferlength is reached, the data in the buffer will be filtered by a low-pass filter: $N(s) = \frac{1}{\frac{1}{2\pi \cdot 10}s+1}$. After filtering a weighting function can be applied to the buffers to make sure the newest data influence the compensation the most. The following weighting function will be used:

$$W(i) = \frac{1}{1 + \exp(-\frac{10}{2 \cdot l} \cdot i)} \tag{5.4}$$

with l the bufferlength and i the bufferelement. This weighting function looks like (Fig. 5.11), choosing a bufferlength of 30000 points for the simulation. Now, the different parameters a, b_1 and

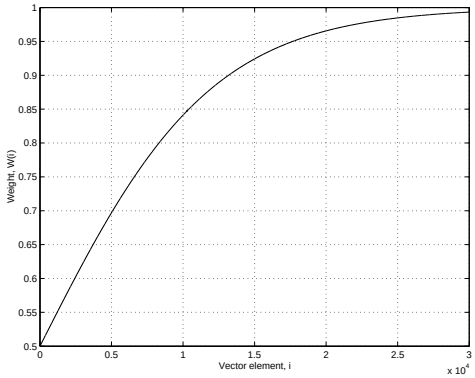


Figure 5.11: weighting function

c can be determined out of the (weighted) buffers: $a = max(UC)$, $b_1 = \frac{2}{max(UV)-min(UV)}$ and $c = b_2 \cdot mean(UY)$. After this, the parameters will be sent to the output of the m-file s-function and the buffers will be cleared. Then, during the following period the computed parameters will be kept and the process repeats itself. Figure B.6 in the appendix shows the implementation into the Simulink model. In this model, a file called 'buffer' is the m-file s-function that contains the buffers, the file called 'function2' is again the c-file that computes the compensation function out of the parameters and the file called 'function3' computes the actual difference between u_v and $\frac{\dot{y}}{s}$ as well as $u_v|_{\dot{y}=0}$. For a more realistic representation of the reality some measurement noise has been applied to the velocity signal. Furthermore, the influence of a temperature change of 15 $[^oC]$ on the viscosity and therefore the density of the oil has been investigated, to see how the algorithm reacts on the small offset that occurs. A simulation has been performed with this algorithm together with the compensation function described by (Eq. 5.3), the results can be seen in (Fig. 5.12) and (Fig. 5.13). Once the buffer reaches its pre-defined bufferlength, the parameters are computed and these are filtered again with a low-pass filter to guarantee a smooth fading-in of the compensation function instead of jerky parameter changes.

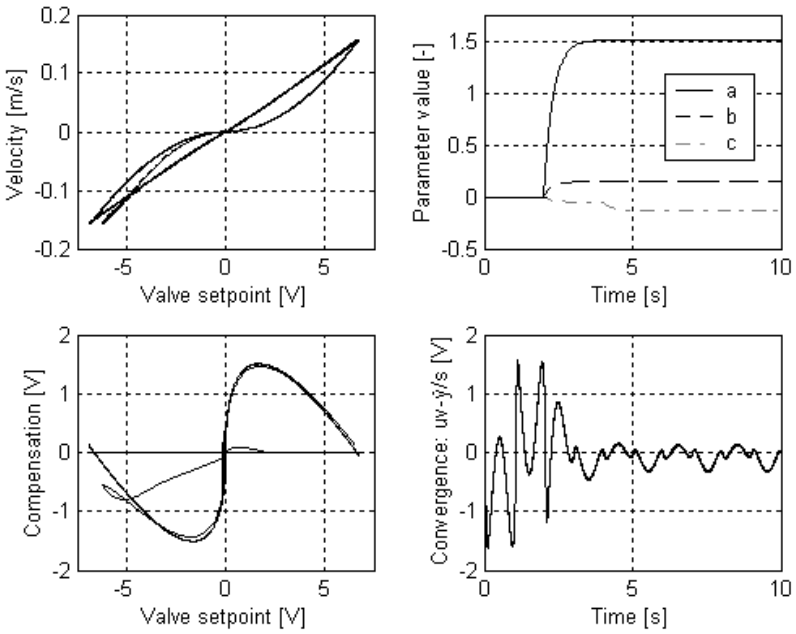


Figure 5.12: simulation results

Figure 5.12 shows the flow-curve, the parameter-evolution, the compensation curve and the difference between u_v and $\frac{\dot{y}}{s}$. The effect of the measurement noise can be seen in the flow-curve and convergence plot, but this noise does not affect the parameter computation as a result of the applied filtering.

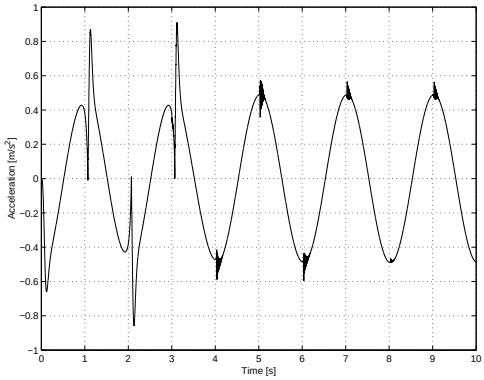


Figure 5.13: acceleration profile

Figure 5.13 shows the acceleration profile of the actuator that results after applying the compensation algorithm as described in this section; the large peaks at start of the simulation disappear as time goes on. As can be seen in (Fig. 5.12), the global shape of the compensation is nearly the right one after the first update. This is because of the fact that there is sufficient data available to calculate reliable parameters. After the first update, only small changes on these parameters will take place. When the system starts the buffers can be filled directly, because the maximum values of the parameters a and b will be kept until larger setpoints are applied to the system. Then, a point on the flow-curve will be

reached for the first time, leading to different values for a and b. Parameter c needs to describe the actual offset from $u_v = 0$, so this value is based on the latest buffer data.

5.6 Discussion

Different types of compensation algorithms have been derived in this chapter. Mainly because of the fact that the important signals like u_v and $\dot{y}$ can contain a certain amount of noise, the choice for a compensation function has been made. First, an iterative approach was designed, using two different algebraic compensation functions. Bounds on the parameter computation must be set to prevent outrageous compensation voltages, using the iterative approach, especially when the required shape of compensation is unknown. This seemed rather risky, therefore the choice was made to look at the *actual* difference between u_v an $\dot{y}$, which prevents unrealistic parameters and leads to the best possible compensation using that particular algebraic function. An algorithm based on parameter computation after storing several signals during a user-definable period has been investigated. This looks like the best method, because the update speed can be chosen, smooth fading-in of the parameters is guaranteed and filtering (together with weighting) is possible over the buffers to be able to compute the most realistic parameters.

Chapter 6

Conclusions and recommendations

The non-linearity in the valve that controls the oil-flow in a hydraulic system is the main cause for smoothness problems. To examine the effects caused by the non-linearity the Micro Motion System is used to indicate the smoothness and perform experiments like pressure and friction measurements to retrieve the required numerical data. Also a friction model has been derived according to a empirical Stribeck-function.

In order to create a simulation model of a hydraulic actuator controlled by a three-way proportional valve several assumptions have been made to come to a set of (differential) equations that describe the different elements. The valve can be divided into three parts, namely its dynamics, its flow-area function and the flow out of the valve. The latter depends on the pressure in the lower cylinder chamber, which can be determined by writing down the equations for the continuity and compressibility of the oil. Finally, the actuator dynamics of the double-acting differential cylinder have been modeled, by looking at the forces acting on the piston (pressures, gravity and friction).

A Simulink-model has been created that can be used for evaluation of the smoothness of a single hydraulic actuator. An important aspect is the choice of flow-area function, which has to be smooth around the mid-position of the valve spool to prevent unrealistic simulation results. The simulation results have been compared to the experimental results and the differences have been acknowledged.

The basic principle in finding the compensation function to cancel the non-linearity in the valve can be determined by looking at the static relationship between valve setpoint and piston velocity when the acceleration and pressure change are both zero. This relation then describes the desired linear relation between valve setpoint and piston velocity, indicated by the flow-curve. For the static case a compensation function can be derived, but the required compensation could vary with temperature changes. Therefore the compensation should be adjusted when the system is moving.

Two different approaches to tackle this problem have been investigated. Because of measurement noise on the different signals, the choice has been made to calculate parameters of an algebraic compensation function out of the shape of the required compensation that is determined iteratively. A great advantage is that filtering can be applied, because then the phase is less important. Two different compensation functions have been investigated, and the simulation results have been shown. When the compensation function is not exactly known in advance, the iterative process may not be the safest choice, due to the fact that it might not converge, which could cause the compensation voltage being too high in which case the smoothness can even become worse.

The main idea behind the second approach was to prevent the problems that occurred in the previous one, so to guarantee the maximum smoothness that can be achieved with a specific compensation function, in a safe way. Hereto, different buffers are used to store the relevant information and once in a user-definable time the parameters are calculated out of the stored data in the buffers. The required magnitude of the compensation voltage is now determined by looking at the actual difference between the valve setpoint and the desired voltage to reach the linear flow-curve. By storing the data in buffers,

also a weighting function can be applied to be able to relate the parameters to the newest information. The simulation results show that this is a safer way to calculate the compensation.

These algorithms have only been tested in numerical simulations. As shown in the differences between the experimental and simulation results, also hysteresis can be present in the real flow-curve. Therefore, a distinction may be required in the compensation voltage for positive and negative accelerations, before testing this algorithm on a real system. If the non-linearity due to dead-band behavior or friction can be determined by looking at the relation between the setpoint and velocity, the second approach can be used for both hydraulic and electrical systems. The algebraic compensation function may have to be adjusted, depending on the kind of linearity.

Appendix A

Numerical data

P_s= $7.0 \cdot 10^6$ $[\frac{N}{m^2}]$
P_t= 0 $[\frac{N}{m^2}]$
C_d= $\frac{\pi}{\pi+2} = 0.61$ $[-]$
ρ= 860 $[\frac{kg}{m^3}]$
g= 9.8 $[\frac{m}{s^2}]$
A_1= $\frac{\pi}{4} \cdot 0.04^2$ $[m^2]$
A_2= $\frac{\pi}{4}(0.04^2 - 0.03^2)$ $[m^2]$
E= $1.5 \cdot 10^9$ $[\frac{N}{m^2}]$
L_s= 0.2 $[m]$
M= 40 $[kg]$
T= 35 $[C]$
$u_{v,max}$=10 $[V]$
$x_{v,max}$=$1.1 \cdot 10^{-3}$ $[m]$
O_l=$5 \cdot 10^{-6}$ $[m]$
d=$1 \cdot 10^{-2}$ $[m]$
V_{pl}= $5 \cdot 10^{-7}$ $[m^3]$
k=100 $[-]$
Stribeck-curve:
σ^+=2200 $[\frac{Ns}{m}]$
F_{c0}^+=65 $[N]$
F_{s0}^+=30 $[N]$
c_s^+=c_s^-=0.005 $[\frac{m}{s}]$
σ^-=1800 $[\frac{Ns}{m}]$
F_{c0}^-=60 $[N]$
F_{s0}^-=30 $[N]$

Appendix B

Simulink (sub)systems

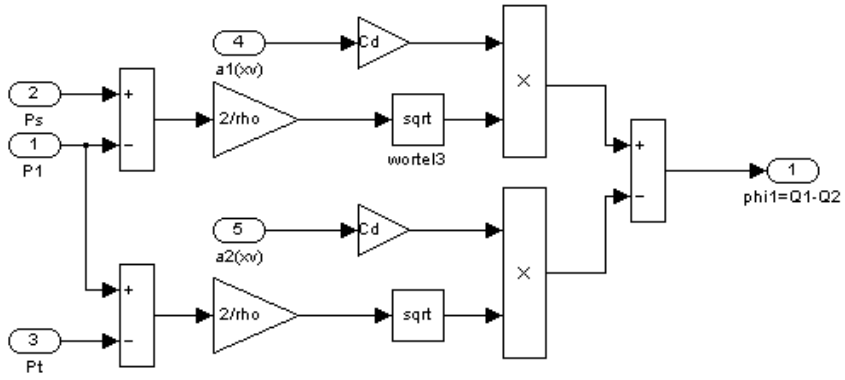


Figure B.1: flow out of valve

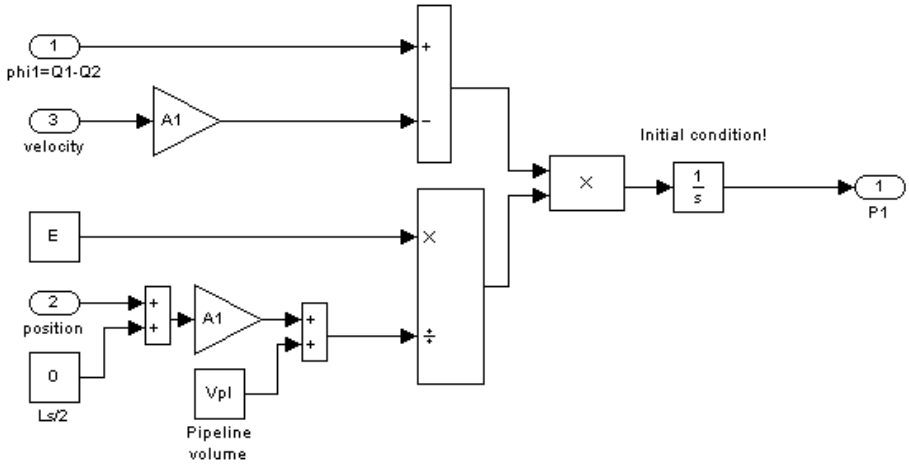


Figure B.2: continuity and compressibility

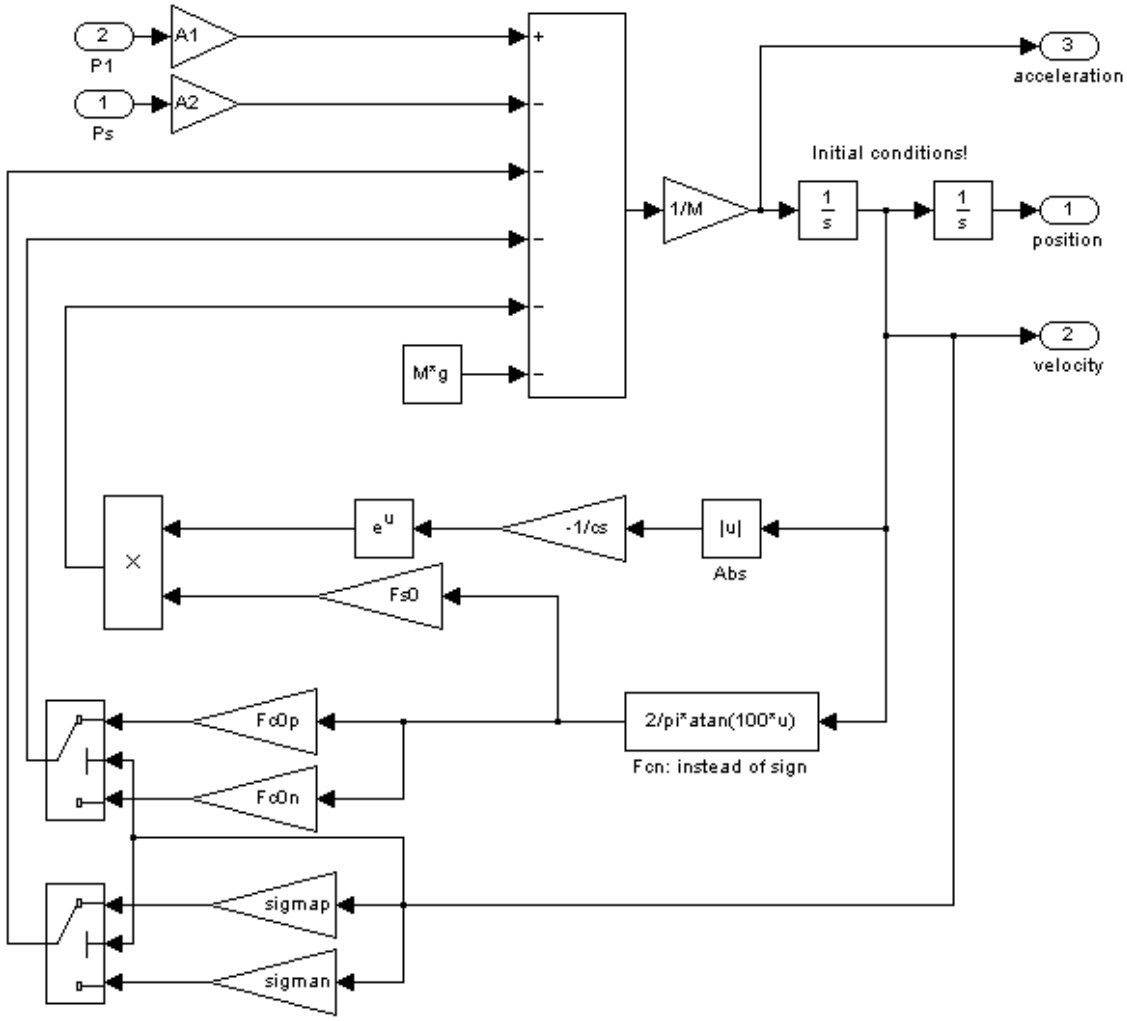


Figure B.3: cylinder

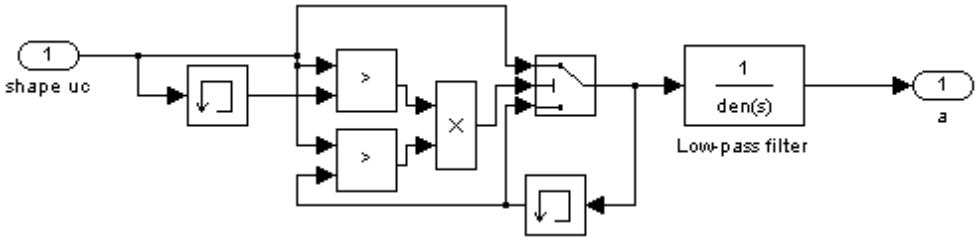


Figure B.4: Simulink block

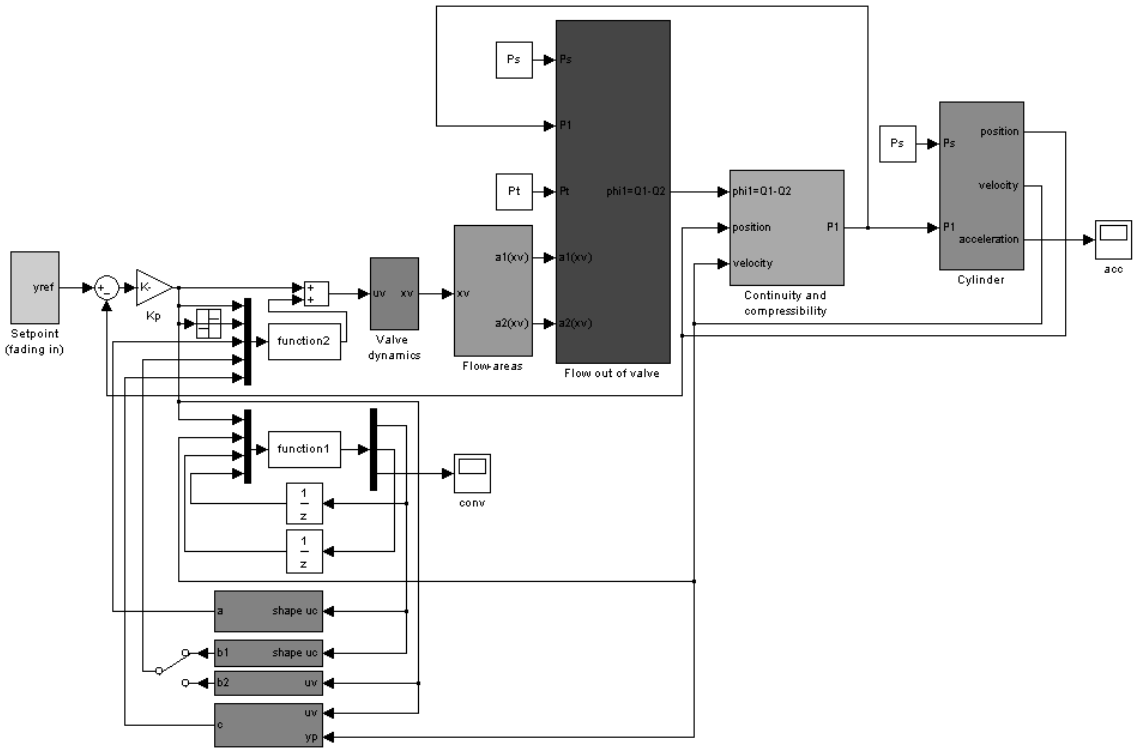


Figure B.5: Simulink model with compensation algorithm (1)

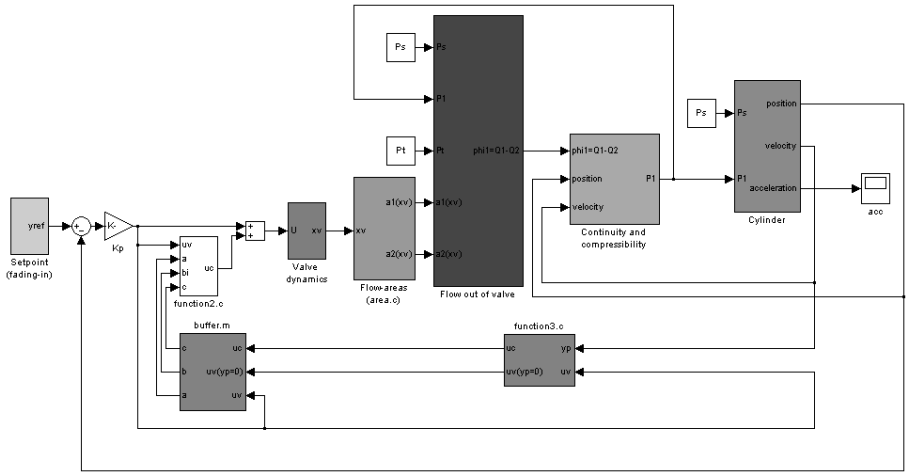


Figure B.6: Simulink model with compensation algorithm (2)

Appendix C

M-file S-function: buffer.m

```
function [y,x0,str,ts] = buffer(t,x,u,flag)

% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage of the S-function.

switch flag,
  %==================================
  % Initialization %
  %==================================
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [y,x0,str,ts]=mdlInitializeSizes;

  %==================================
  % Outputs %
  %==================================
  % Return the outputs of the S-function block.
  case 3
    y=mdlOutputs(t,x,u);

  %==================================
  % Unhandled flags %
  %==================================
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    y=[];

  %==================================
  % Unexpected flags (error handling)%
  %==================================
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end     % end buffer
```

```
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================

function [y,x0,str,ts] = mdlInitializeSizes()

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 3;
sizes.NumInputs      = 3;
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

y = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes

%=============================================================================
% mdlOutputs
% Return the output vector for the S-function
%=============================================================================

function y = mdlOutputs(t,x,u)
global UV UC UY    % initial values: UV=UC=UY=[]; and anew,aold,b,c,q=0; in workspace
global anew aold bnew bold c q

if (length(UV)<30000)
    UV=[UV u(1)];
    UC=[UC u(2)];
    if(u(3)~=0.0)
    UY=[UY u(3)];
    end
    y(1)=aold;
    y(2)=bold;
    y(3)=c;
else
    % discrete filter %
    NUM=[0.003137];
    DEN=[1 -0.9969];
    UVF=filter(NUM,DEN,UV);
    UCF=filter(NUM,DEN,UC);
    UYF=filter(NUM,DEN,UY);
    for i=1:30000,
        w=1./(1+exp((10./30000).*-i));
        UCF(i)=w.*UCF(i);
        UVF(i)=w.*UVF(i);
    end
    q=(max(UVF)-min(UVF))/2;
```

```
    q=max(abs(UVF));
    bnew=1/q;
    anew=max(UCF);
    c=b*mean(UYF);
    if (anew>aold)
        y(1)=anew;
        aold=anew;
    else
        y(1)=aold;
    end
    if (bnew>bold)
        y(2)=bnew;
        bold=bnew;
    else
        y(2)=bold;
    end

    y(3)=c;
    UV=[];
    UC=[];
    UY=[];
end

% end mdlOutputs
```

Bibliography

[1] P. Beater. *Entwurf hydraulischer Maschinen: Modellbildung, Stabilitaetsanalyse und Simulation hydrostatischer Antriebe und Steuerungen*. Springer, 1999.

[2] Mohieddine Jelali and Andreas Kroll. *Hydraulic servo systems, modeling, identification and control*. Springer-Verlag London Limited, 2003.

[3] R.I. Leine. *Bifurcations in discontinuous mechanical systems of Filippov-type (proefschrift)*. Universiteitsdrukkerij TU Eindhoven, The Netherlands, 2000.

[4] Nissing D. Polzer, J. Trajectory adaptation for flatness based tracking and vibration control on a flexible robot. In *Proc UKACC Int Confer Control 2000, Cambridge, UK*, 2000.

[5] Rexroth Hydraudyne. *Operation Manual 6 D.O.F. Micro Motion System*, 2000.

[6] R. Stribeck. *Die wesentlichen Eigenschaften der Gleit- und Rollenlager*. Z Ver Dtsch Ing, 1902.

[7] F.D. Stuyvenberg and J.Vinke. *Hydrauliek*. Educaboek B.V., 1972.

[8] Taco J. Viersma. *Analysis, synthesis and design of hydraulic servosystems and pipelines*. Elsevier scientific publishing company Amsterdam, 1980.

[9] T.J. Viersma and P.C. Teerhuis. *Inleiding op hydraulische besturingen*, 1991. Diktaat i29A, TU Delft.