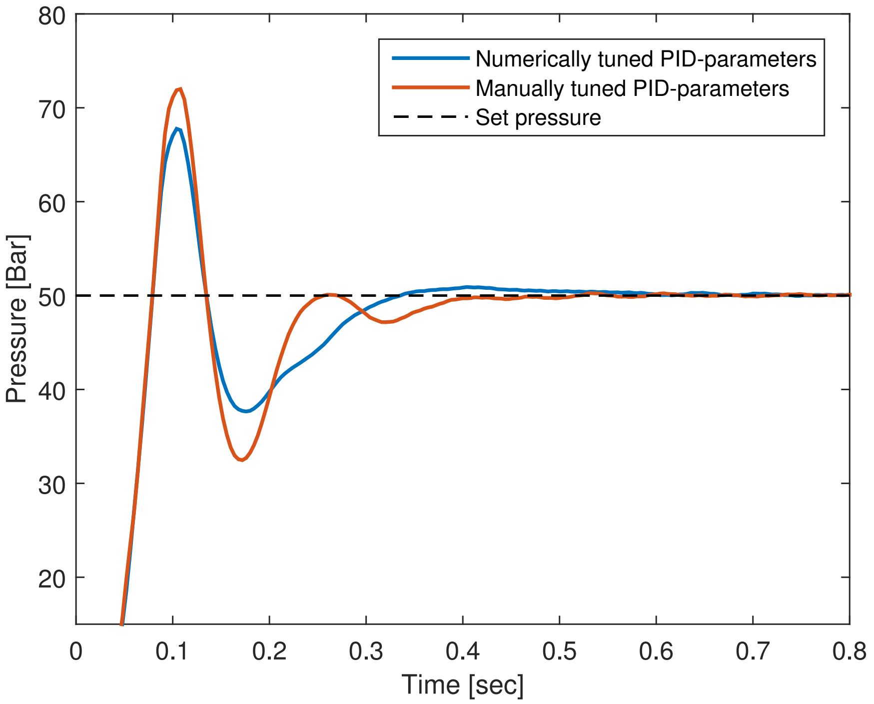
CHALMERS
UNIVERSITY OF TECHNOLOGY



Numerical Tuning of Control Parameters in Hydraulic System

Evaluation of System Identification as modeling method for PID-controller tuning

Master's Thesis in Systems, Control and Mechatronics

MARCUS GRÖNBÄCK

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Master's Thesis 2015:NN

Numerical Tuning of Control Parameters in Hydraulic System

Evaluation of System Identification as modeling method for PID-controller tuning

MARCUS GRÖNBÄCK

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden 2015

Numerical Tuning of Control Parameters in Hydraulic System
Evaluation of System Identification as modeling method for PID-controller tuning
MARCUS GRÖNBÄCK

Supervisor at AP&T: **Anders Kullberg**, *Electrical Systems and Software Design Manager, AP&T Sweden AB*
Supervisor at Chalmers: **Jonas Sjöberg**, *Department of Signals and Systems, Chalmers University of Technology*
Examiner: **Jonas Sjöberg**, *Department of Signals and Systems, Chalmers University of Technology*

Master's Thesis 2015:NN
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Comparison of performance between manually tuned and numerically tuned PID-controller.

Typeset in LaTeX
Gothenburg, Sweden 2015

Numerical Tuning of Control Parameters in Hydraulic System
Evaluation of System Identification as modeling method for PID-controller tuning
MARCUS GRÖNBÄCK
Department of Signals and Systems
Chalmers University of Technology

Abstract

This thesis presents the development of a numerical PID-controller tuning tool for a hydraulic system. The hydraulic system consist of proportional pressure valve that control the blank holding pressure of a hydraulic cylinder. A model based design methodology has been used where a model of the hydraulics system has been created using blackbox system identification. The estimated model has been used to numerically tune the PID-parameter for the regulator controlling the proportional pressure valve. The numerical tuning improved the performance of the controller compared to the previously used manual tuning of the P, I and D parameter. Mathwork´s *System Identification Toolbox* and *Simulink Control Design* has been used to develop a framework for quickly estimate a valid model from input-output data and numerically calculate the PID-parameters.

Keywords: System Identification, PID-Controller, Modeling, Tuning, Hydraulic, Press

Preface

This master thesis was written in order to complete the studies at the master program Systems, Control and Mechatronics at Chalmers University of Technology. The thesis has been written at AP&T AB in Tranemo during the spring of 2015 and covers 30 credits.

Acknowledgements

I would like to thanks everyone involved in this project. Especially my colleagues at AP&T for there friendly and helpful attitude. An extra thanks to Anders, David, Masoud, Per, Thomas, Magnus, Daniel and Christer at AP&T, without you this thesis would not be possible. I also would like to thanks my supervisor and examiner Jonas for your guidance during the project. Last but not least I would like to thanks my family for your hospitality and support during the weeks and my fiancée Mimmie for your patience all the time I have been away from you during this project.

Marcus Grönbäck, Tranemo, May 2015

Contents

1

Introduction

This thesis will present the development of a numerical PID-controller tuning framework designed to be used during the start-up process of a hydraulic press. Model based design is used to create a model of an existing hydraulic system and numerically find acceptable parameter settings for the PID-regulator controlling the system. The hydraulic system is a hydraulic outlet of a hydraulic press connected to a hydraulic cylinder. The hydraulic outlet has a proportional pressure valve which will be controlled. The hydraulic press is designed and built by AP&T Sweden AB.

1.1 Background

The quality of the machines and the ability of making money are two important factors in industry today. In a global market the companies constantly need to improve to survive the competition. An example of improving the design and construction process is to work with model based design. In model based design the product is design using computerised models of the real system. When a model is created the regulators controlling the system can be tuned numerically to get acceptable parameters for the modeled system. The tuning of controllers are crucial to achieve the best possible performance of the system.

At AP&T, the tuning of the controllers are done manually using a trial and error method. The manual tuning is a time consuming process and the start-up operator performing the manual tuning do not know when to settle with the tuning. Since there are several start-up operators performing the manual tuning the performance of the controller differ depending on which operator performing the tuning. A new, better and fast method for the controller tuning is needed which will improve the performance of the machine and make all start-up operators tune the controllers in the same way.

No previous work has been done at AP&T in the area this thesis cover. AP&T do not have any mathematical model of hydraulic system and do not use model based design in the design phase of the control system. Nevertheless, they have good system knowledge from previous project which they use when designing the machines.

1.2 Scope

The scope of this thesis is to evaluate black box system identification for building a plant model and investigate if the numerical tuning tool can be developed using this modeling method. The system identification process will include data collection and estimation of a model from the collected data. The control structure of the hydraulic press needs to be investigated and recreated in a simulation environment. The numerical tuning of controller will be based on the control structure and the plant model.

The function of the hydraulic cylinder covered in this thesis is the blank holding pressure. The blank holding pressure is the function when the hydraulic cylinder is exposed to an external force and holds the pressure in the cylinder at the set point value.

1.3 Requirements

The numerical tuning framework requires a quick and reliable way of estimating the plant model. The collected data from the machine will be exported to a desktop PC running MATLAB where the model building and tuning will be done. The plant model needs to be stable, casual, minimum phase, controllable and have a fit of at least 85 % against validation data. The control structure need to be verified to insure a correct implementation. Close loop data from the real machine will be compared with simulated close loop data to verify implementation. The numerical tuning framework will only be designed for blank holding pressure with a proportional pressure valve. The manual tuning of a controller is estimated by the start-up operator D Staafjord 28 May 2015 to 30 minutes. The numerical tuning tool will be designed to take a maximum of 10 minutes of the start-up operators time.

1.4 Contribution

The contribution of this thesis is the development of a numerical PID-controller tuning framework. The numerical tuning results in a proof-of-concept where a controller is tuned with a improved performance compared with manual tuning. The thesis results in a large step towards numerical tuning of the PID-regulator controlling the proportional pressure valve during blank holding. With a fully implemented numerical tuning software the lead time would be shorten and the performance of the controllers would be improved.

1.5 Software

The main software used were MATLAB and Simulink. The system identification has been performed using the MATLAB toolbox *System Identification Toolbox*, described in [9]. The control structure has been implemented using Simulink and the tuning of controllers has been done using the Simulink toolbox *Simulink Control Design*, described in [10]. The data collection has been done with the PLC software *Siemens Simotion Scout*.

1.6 Thesis Outline

The outline of the thesis is arranged as follows:

Chapter 1 - Introduction including background, scope, requirements and contribution for the thesis. The company presentation is also included in the chapter.

Chapter 2 - Description of the analyzed hydraulic system and control system.

Chapter 3 - The model estimation using System Identification is covered.

Chapter 4 - The control structures implementation in Simulink is presented.

Chapter 5 - Details of the PID-parameter tuning procedure.

Chapter 6 - Results of the tuned PID-parameters tested on a real machine.

Chapter 7 - Discussion of results, future work and conclusion are covered.

1.7 Company Presentation

The thesis work was carried out at AP&T AB in Tranemo, Sweden. AP&T is a company designing, producing and selling customer designed manufacturing lines for metal forming. AP&T design and build hydraulic presses, robots placing the metal part in the press as well as the tools used to form the metal. AP&T sells production lines to customers all over the world. AP&T is in an expansive phase and need to take every part of the company to the next level to reach the high set goals.

Figure 1.1: Example of press hardening production line sold by AP&T.

1.7.1 Press Hardening

AP&T is world-leading in designing and producing press hardening production lines. Press hardening is a technique producing lightweight high tensile steel. The steel is first heated to 950 °C. The hot steel is then formed by a press and at the same time cold down. This procedure creates a new allotrope of the steel with a lower weight and higher tensile strength. According to [4] a mid-sized car can reduce the weight of the steel with 68 kg if press hardened steel are used instead of the commonly used cold formed metal. The weight reduction result in a reduction of fuel consumption with about 0.1 liter per 100 km according to [4]. Due to the high tensile strength press hardened parts are commonly used in A- and B-pillars in cars which improve the safety. An example of a car using press hardened A- and B-pillars produced in AP&T production lines are the new Volvo XC90. AP&T has developed a complete manufacturing line producing press hardening part. The line consist of an oven heating the metal, robots moving the metal part and a hydraulic press with press tools simultaneously forming and cooling down the steel. In Figure 1.1 an example of a press hardening production line can be seen.

2

System Description

This chapter includes a description of the analyzed system. The hydraulic components will be presented as well as the control system of the hydraulic press.

A hydraulic press consist of many hydraulic subsystems e.g slide, die cushion and a number of hydraulic outlets. It is essential to control the pressure in all these subsystems. Two different ways of controlling the pressure are used at AP&T, load sensing and pump controlled. In load sensing the pumps provides a high enough pressure and a proportional pressure valve controls the pressure in the cylinder. In pump controlled the system pressure is controlled directly by the pump controller that start and stops the pump. Tuning of the pump controller parameters are rarely done by a AP&T start-up operators but are taken care of by the supplier of pumps. The load sensing system is however tuned manually by the start-up operator and therefore this thesis will focus on this tuning process. For every hydraulic outlet, third cylinder, die cushion cylinder etc. there is a proportional pressure valve with a PID-controller that need to be tuned by the start-up operator.

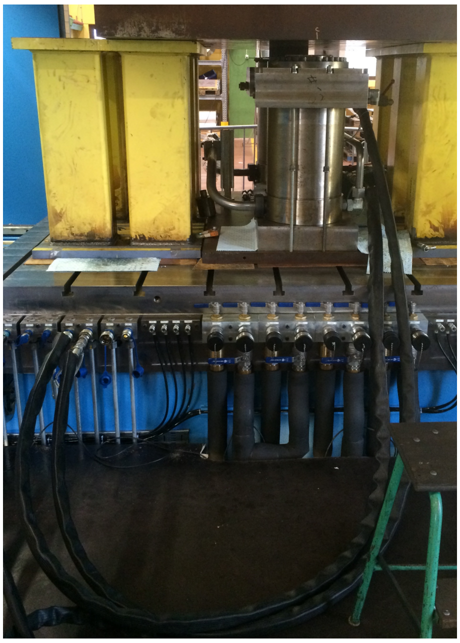


Figure 2.1: Picture of the hydraulic outlet connected to the hydraulic cylinder.

The machine used for data collection was a press hardening press with a press force of 12 000 kN. The slide was pump controlled and there was no die cushion in this design. However, there were four load sensing hydraulic outlets which were decided to be the system to be analyzed. Hydraulic outlets are mainly used in the press die. A hydraulic outlet is basically an outlet where a hydraulic cylinder can be connected, see Figure 2.1. A proportional pressure valve is present for every outlet which control the pressure in the system. The hydraulic cylinder can have three possible applications; move in one or both directions and blank holding at a certain pressure. The action considered in this thesis is the blank holding pressure.

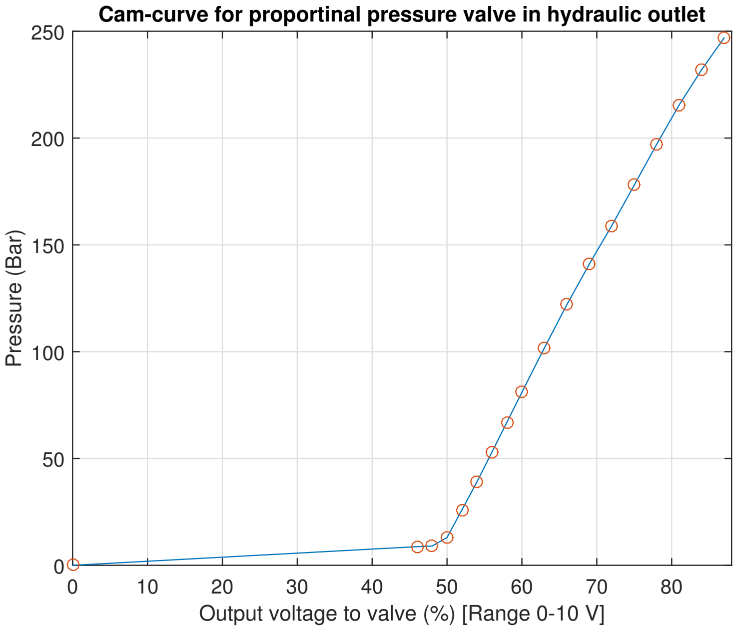


Figure 2.2: Example of CAM curve for a proportional pressure valve.

The system consist of an electric motor which drives a hydraulic pump. The pump loads the accumulator with a working pressure of approximately 220 bar. A proportional pressure valve is controlled by the PLC where a PID-regulator control the valve input signal depending of the set and actual pressure. The input signal to the valve is a voltage between 0-10 V. There is also a proportional directional valve that control the direction of oil flow. A double acting hydraulic cylinder is connected to both valves. In the system there is a pressure transducer on the plus side of the hydraulic cylinder. The measured value is fed into the controller block in order to control the pressure. There is also a pressure transducer after the accumulator to insure supply pressure. The system is designed to handle pressures between 30-220 bar. All pressure transducers are low pass filtered with a time constant of 12 ms. A sketch of the analyzed system can be seen in Figure 2.3.

2.1 Cam-Curve

To translate the valve input signal in volts to the desired pressure a function relating these proprieties are created. This function is called a CAM-curve. One CAM-curve is created for every proportional pressure valve since their properties may vary. A CAM-curve is a way of linearize the valve. The valve is typically nonlinear but the static performance is given in the CAM-curve. By setting a input voltage to the valve and measuring the obtained pressure a curve describing the characteristics of the valve is given. An example of a CAM-curve for a proportional pressure valve can be seen in Figure 2.2.

When a CAM-curve is created the slide is pushing down the cylinder with a constant speed. It can be argued that the CAM-curve will look different if a higher or lower speed of the slide is used. The argument is true, but since the curve do not radically change the CAM-curve is only measured once with a normal working speed for the slide, which usually is 90 mm/s.

2.2 Control System

The control system used in the press is a Siemens Simotion PLC. The PLC has a loop time of 2 ms and the code is written in structure text. In Figure 2.4 a block diagram of the PID-controller can be seen. The controller code is provided by the supplier of the PLC and can not be modified.

All transducers in the machine are filtered to avoid high frequency noise. The mathematical expression of the low pass filter can be seen in (2.1), where u is the input and y the output. For every scan cycle the current values of u and y are saved to $uOld$ and $yOld$ to be used in the next cycle.

$$y = y_{Old} + ((u + u_{Old}) * 0.5 - y_{Old}) * (1 - \exp(-CycleTime/FilterTime)) \tag{2.1}$$

$CycleTime$ is always set to the scan cycle time which in this machine is 2 ms. The $FilterTime$ has to be set higher than the $CycleTime$. The $FilterTime$ is possible to change during the start-up of the press but this is rarely done and the default value of 12 ms is normally kept.

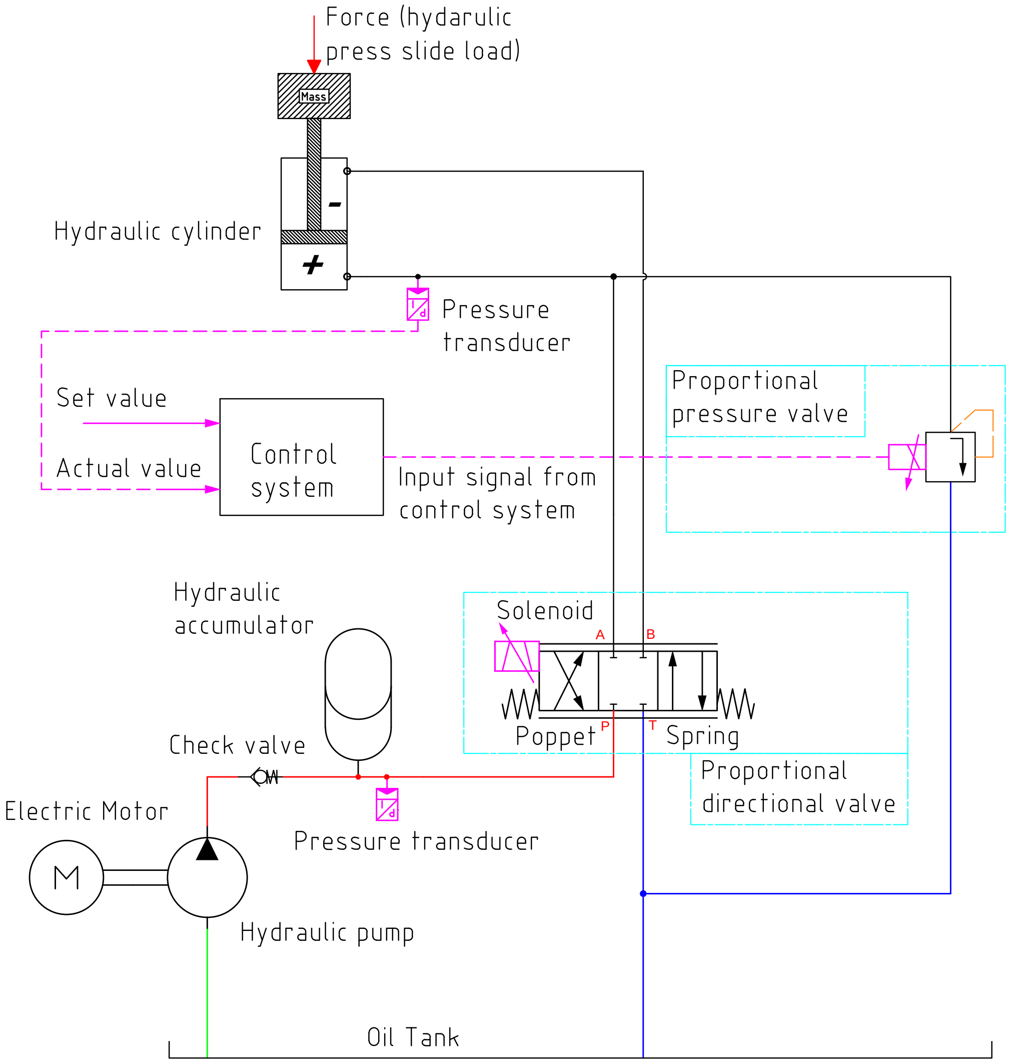


Figure 2.3: A graphical visualisation of the analyzed hydraulic system.

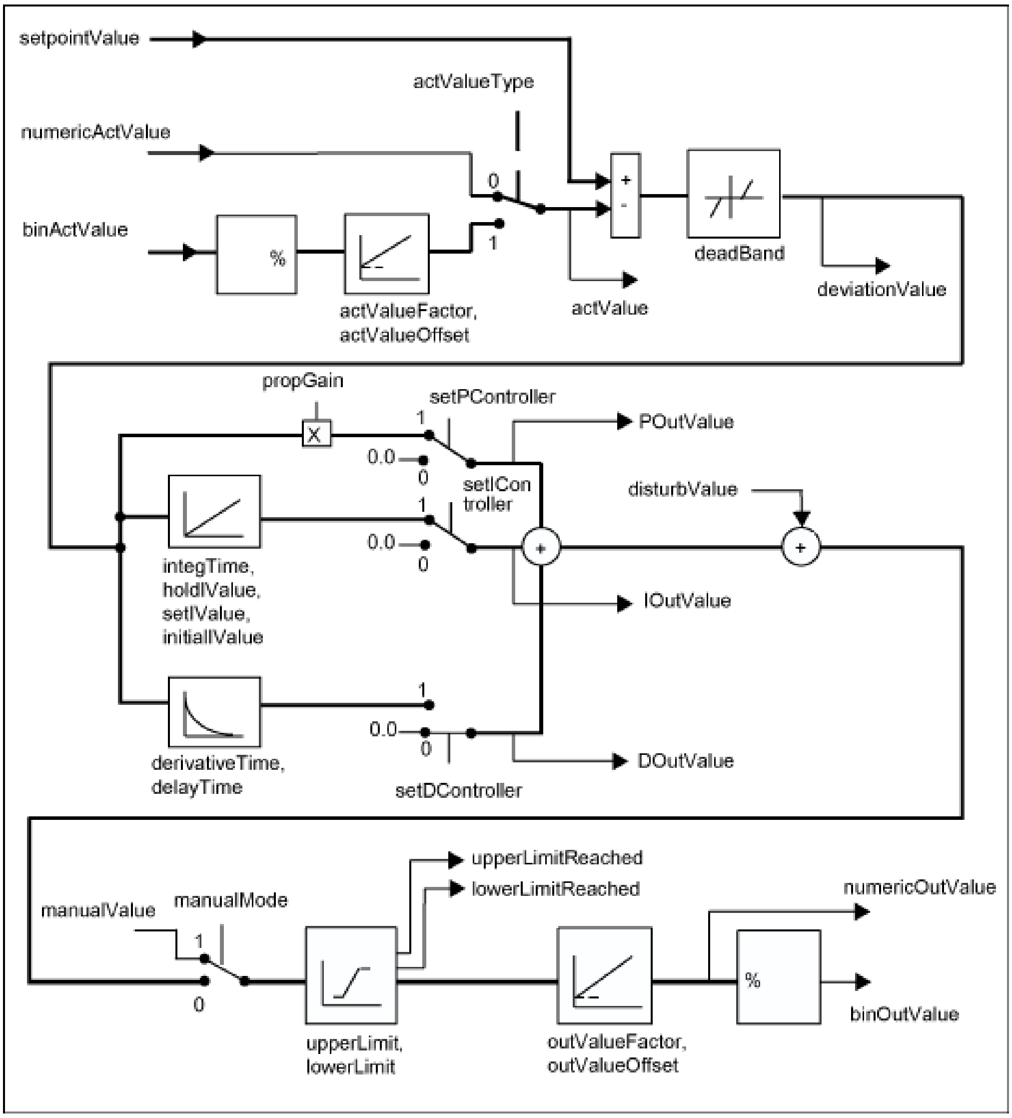


Figure 2.4: Block diagram of the PID-controller used in the PLC.

3

System Identification

This chapter describes the modeling of the system described in Chapter 2. According to Ljung and Glad [8] two different ways of building models are present; *physical modeling* and *identification*. In this thesis, system identification was selected for modeling the system. This method was used due to the fact that the real machine was available for experiments and data collection.

Data driven modeling like system identification is suitable when the dynamics of the system are too complex to derive the complete physical model. System identification is also less expensive due to the complex and time demanding task to create a valid physical model. By measuring the input and output signals, the dynamics of the system can be estimated using a system identification software. The success of identification modeling highly depend on the quality and information in the collected data. Informative set of data gives a higher probability of estimating the dynamics of the system in the best possible way. It is almost impossible to find an exact model that maps one-to-one to reality but the modeler need to settle with an estimation that mimic the system "good enough". For further information about system identification, see [1].

The data has been collected without any controller active (open loop) to find out the dynamics of the system. When the system is in open loop the input signal to the valve is taken directly from the value in the CAM-curve. For example, if the input signal for the blank holding pressure is 100 bar the valve will constantly be fed with approximately 6.2 V using the CAM-curve in Figure 2.2. The model is created with the purpose to control it and calculated the PID-parameters for the modeled system.

3.1 Data Collection

The system that should be analyzed was the blank holding pressure of a hydraulic outlet. The blank holding pressure means to resist an external force with a certain pressure. The external force used in this experiment was the slide. The force from the slide was much larger then the force from the cylinder due to the large difference

(a) The slide free falls until this position. From this position the slide hold 90 mm/s.

(b) The slide pushing the cylinder connected to the hydraulic outlet down and the data collection starts.

(c) The slide is almost down at the distances and the data collection stops.

Figure 3.1: Picture of the sequence where the data was collected. This sequence was repeated 10 times for each input signal.

in area. This fact gave the assumption that the velocity of the slide was constant.

The hydraulic cylinder used was 300 mm long and the velocity of the slide was set to 90 mm/s which is considered as a normal working velocity according to start-up operator D. Magnusson 4 Mar. 2015. This gave a total time where the data could be collected to 3.3 seconds. However, some data point in the beginning and the end of the cycle was cut out to minimize the risk of getting some bad data. The real collection time of data was about 2.7 seconds for each stroke. With a sampling time of 2 ms it gave 1350 sample for each stroke. All data has been collected with the hydraulic oil at 45 °C which is normal working temperature according to D. Magnusson 4 Mar. 2015. In Figure 3.1 pictures from the data collection can be seen.

3.1.1 Input Signals and Preparation of Data

The design of input signal is an important decision. Preferably, a couple of different input signals could be evaluated since the dynamics of the system is unknown. A commonly used signal, especially on linear systems are the pseudo random binary signal (PRBS). An example of this signal can be seen in Figure 3.2.

However, experience of identification of hydraulic system shows that PRB-signals might not be a suitable input signal. A wiser choice is a PRM-signal (Pseduo random multilevel signal), see Figure 3.3, which is more suitable at nonlinear systems. Jelali and Kroll [3] state that hydraulic systems in general are considered to be nonlinear. This statement is based on the compressible hydraulic fluid, friction in pipes and in the hydraulic cylinder as well as the complex flow properties of the valves. The same statement was also found in [5].

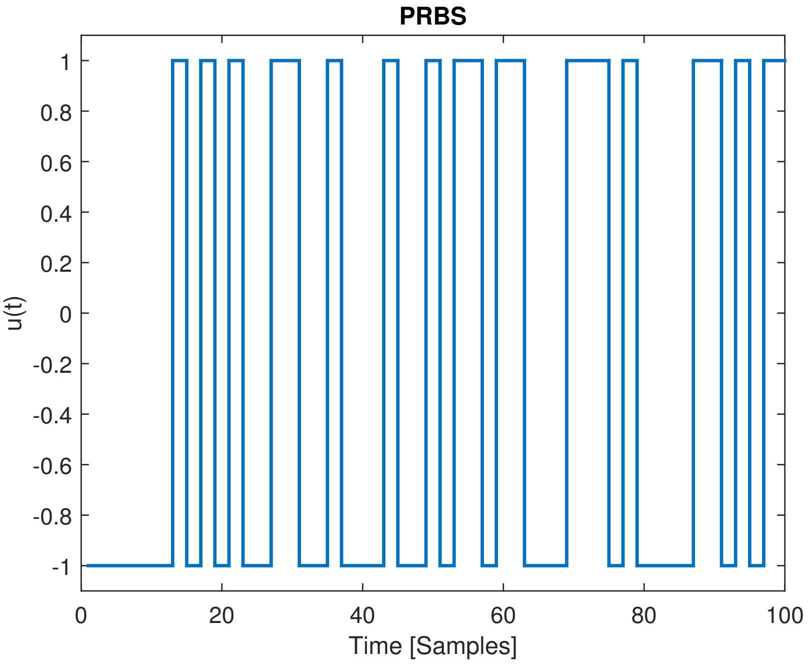


Figure 3.2: Example of pseudo random binary signal. The signal change between two levels of amplitude.

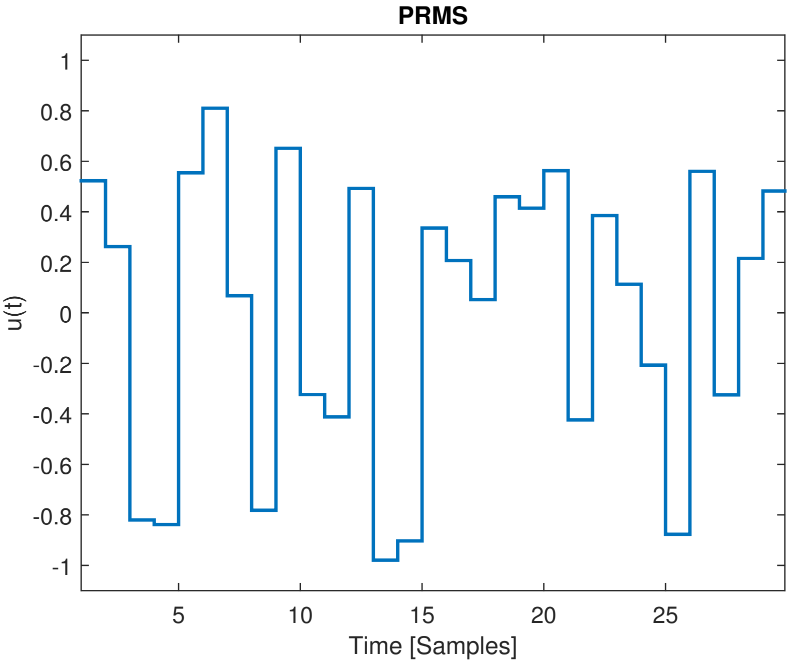


Figure 3.3: Example of pseudo random multilevel signal. The signal change between various levels of amplitude.

Two types of signals were used in the data collection, pseduo random multilevel signal (PRMS) and multifrequency sinusoidal signals. All data was collected in the same machine with the same oil temperature. For every input signal at least 10 strokes were done in order to find disturbance signals and study repeatability. All input signals used in the identification experiment can be seen in Appendix A. When the data is collected Ljung [1] recommend to visually analyze the data both in time and frequency domain. The visual analysis might find some part of the data invalid. These are called outliers and should be removed since they will influence the model incorrectly. If the data consists of many outliers the data collection should be done again to get a better set of data. As can be seen in Appendix A the data in time domain looks good and no outliers were found. The same result was given when the data were analyzed in frequency domain.

The levels of the PRM-signals were randomly produced and the time at each step was set close to the settling time. The amplitude was set inside the working range of the hydraulic outlet which is between 30-220 bar.

The PRMS data has been used to estimate models of the system since steps between different levels are a common application area for the hydraulic outlet. The multi frequency sine signals has been used to analyze system behavior e.g., if the system is an LTI system or not. The data has been split up into estimation data and validation data. All the odd numbered strokes has been merged into estimation data and all even strokes has been merged to validation data. Since three different PRMS has been used there are totally 15 different experiments in each data file. The reason for doing multiple data collections with different input signals were to find an input signal that trigger important frequencies in the system necessary for creating an acceptable model.

The input signals were coded in structured text and place in a PLC routine that executed every scan cycle with a cycle time of 2 ms which corresponds to a sampling frequency of 500 Hz. All pressure transducers were low pass filtered with a filter time of 12 ms. This corresponds to a bandpass frequency of 83 Hz. This low pass filter also avoid aliasing since the bandpass frequency is lower than the sampling frequency [3].

It is recommended to remove mean and scaling of the input and output data before starting estimating linear models. The estimation algorithm gets numerically more robust and gives generally better estimated models with removed mean and scaling [3]. However, if nonlinear models should be estimated the trend and mean should be unchanged since the nonlinearity in the model handle this aspect, for more details see [11]. Since one of the investigated areas were if the system should be estimated with linear or nonlinear models the decision was made to not remove mean and trend of the collected data.

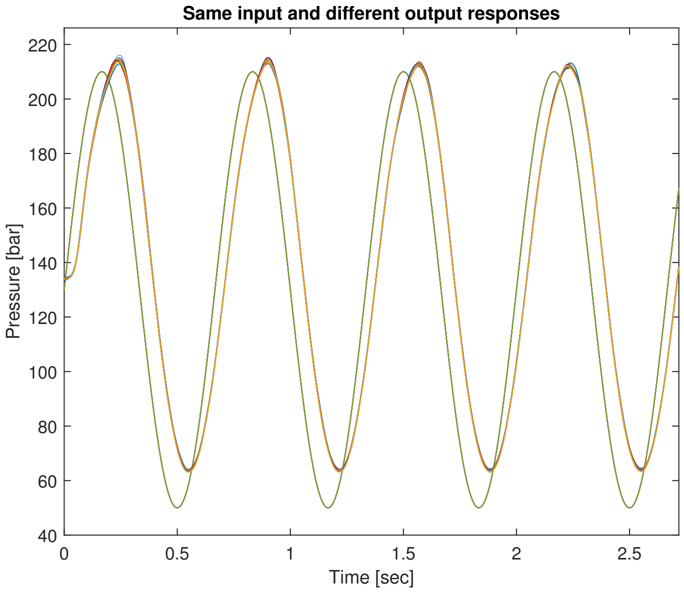


Figure 3.4: Input-output data of single frequency sine with frequency of 1.5 Hz. 10 different strokes placed on top of each other.

3.2 Linear or Nonlinear Model Structure

To investigate of the analyzed hydraulic system should be estimated using linear or nonlinear model structures two different test were performed.

3.2.1 LTI-System

If the system should be considered as a linear time invariant (LTI) system only the amplitude and phase should change from the input to the output signal. If the frequency of the output signal differ from the input signal the system can not be considered an LTI-system.

To get a hint about the system properties an experiment was done where a single and multi frequency sine signal was applied to the system. In Figure 3.4 the experiment data of the single frequency sine signal can be seen. Both the input and output signals are plotted in the same figure. Totally 10 strokes was done and these are place over each other to get a hit about the repeatability as well. In Figure 3.5 the fast fourier transform (FFT) of the data in Figure 3.4 can be seen.

As can be seen in Figure 3.4 and 3.5 only the amplitude and phase change. The same result is obtained for the multi frequency sine signals found in Figures B.4, B.5 and B.6 in Appendix B. The frequency remain the same which indicate that the system can be considered an LTI system.

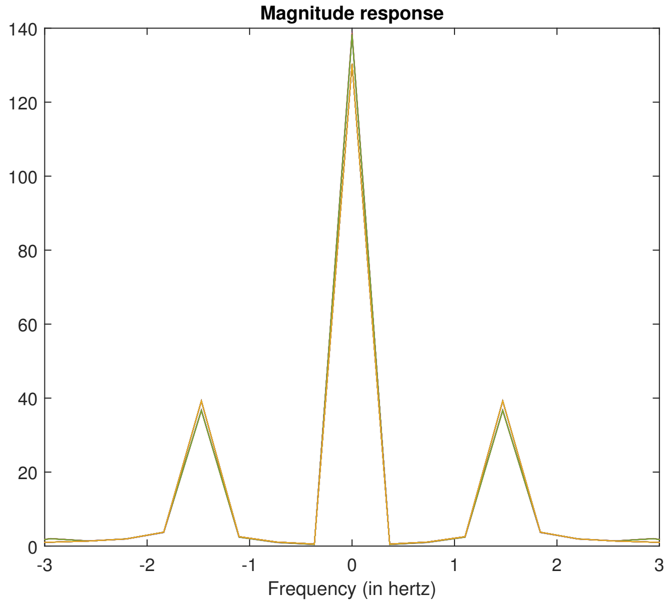


Figure 3.5: FFT of the collected input-output data seen in Figure 3.4.

3.2.2 Different Working Points

Another method of investigate if the system is linear or not is to visually investigate the step response at different working points. A linear system has similar dynamics at all working points. In an experiment one low (60-75 bar) and one high (185-200 bar) working point was selected to be analyzed. The open-loop data was collected in the same machine, with the same oil temperature and the same CAM-curve described in Chapter 2. In Figure 3.6 the result of the data collection can be seen. To visualize more clearly the high working point has been scaled down to the same level as the low working point which can be seen in Figure 3.7.

As can be seen in Figure 3.7 the step responses look quite similar in the shape. The high working point has a little higher starting pressure which remains through the whole experiment. The reason for this according to D. Magnusson 4 Mar. 2015 might be that the valve is better suited for lower pressures and thereby more precise in these regions when the system is in open loop. This result shows that the step responses at two different working points shows an similar dynamics which would indicate a linear model estimation could be possible.

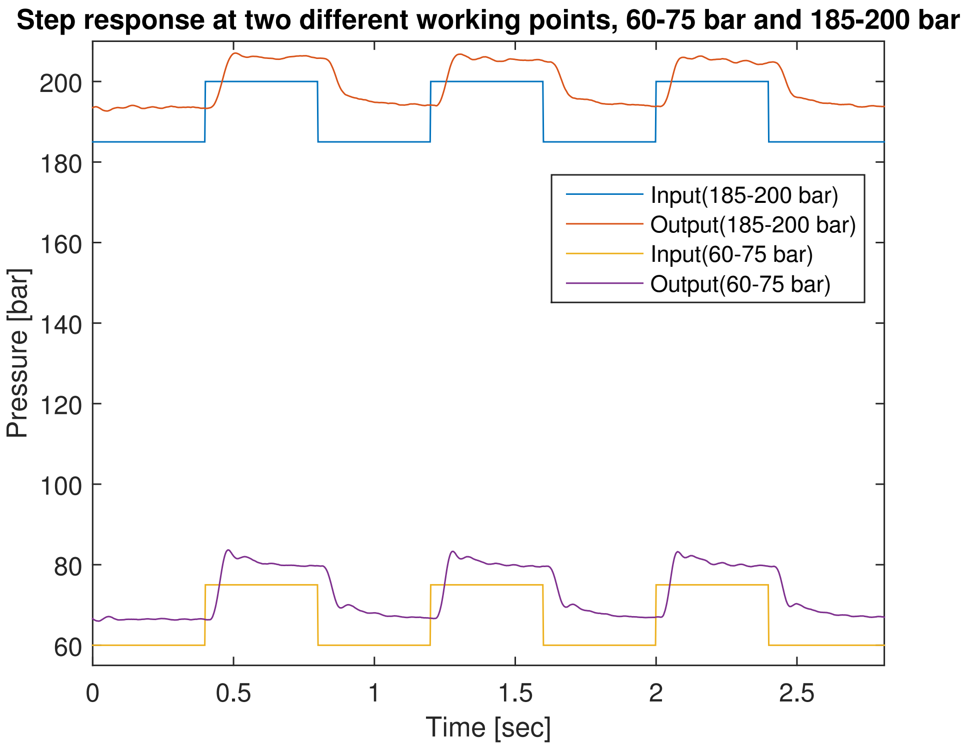


Figure 3.6: Collected data of one stroke of the high and low working point.

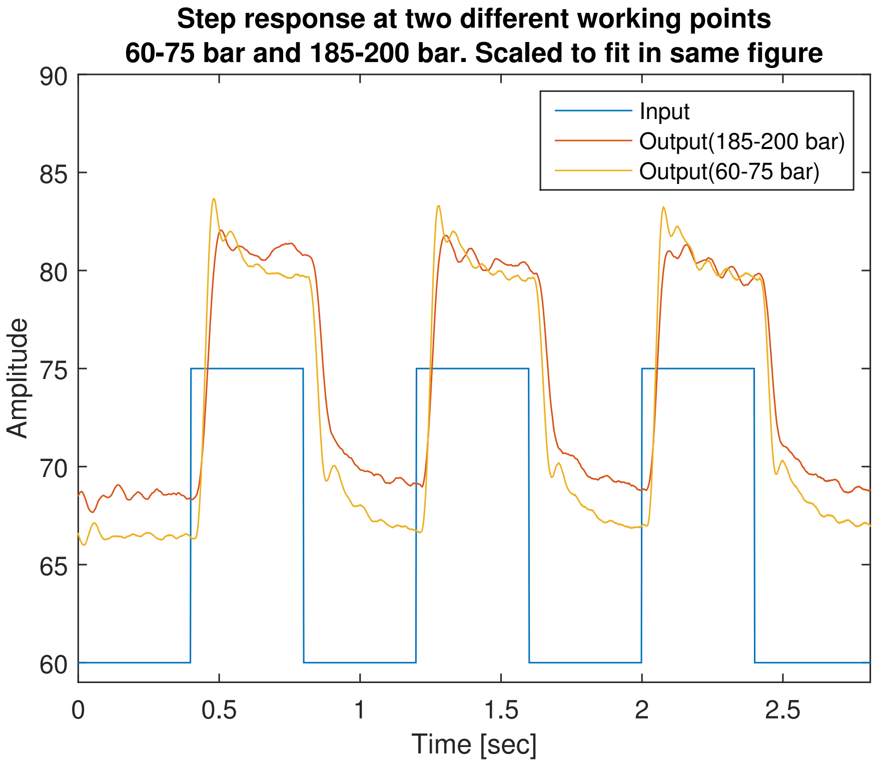


Figure 3.7: The high working point has been scaled down with a factor of 125 to fit to the low working point.

3.3 Model Structure

One of the most important steps in identification is deciding upon a good model structure for the estimated system. Jelali and Kroll [3] describes the estimation of the model fairly straight forward if the model structure is given. However, if the model structure is poorly chosen it is almost impossible to produce an acceptable model.

With the results found in the experiments in Section 3.2 some hints has been given that linear model estimation might be enough for the system. Even if the hydraulic system with a high probability includes nonlinearities it is a good idea to follow the parsimony principle, described in [2], when creating a black box model and start with an easy model structure and gradually increase the order. The aim is to capture the dynamics of the system with the easiest model structure with the lowest order. Previous work has been found where nonlinear hydraulic system has been successfully estimated using linear model structures, see [6] and [7].

Three different model structures has been investigated but also one nonlinear model structure to be able to compare the performance between linear and nonlinear models. The linear model structures investigated were *Output Error* (OE), *AutoRegressive with eXternal input* (ARX) and *AutoRegressive Moving Average models with eXternal input* (ARMAX). Hammerstein-Wiener was the nonlinear model structure investigated. For all model structures an input-output delay should be sent as parameter to the estimation function. This delay was estimated with the MATLAB function *delayest*, described in [12].

A script was developed that test different permutation of the model structure to find out suitable values for the changeable parameters which were the order of the polynomials in the estimated model. For all estimated models a test was done to find out if the model met some predefined requirements:

- All poles inside the unit circle. (Stable model)
- The number of poles should be larger than the number of zeros. (Casual system)
- All zeros inside the unit circle. (Minimum phase)

If all these requirements were met the model was saved. Of all the saved models the one with the lowest order but still with good fit was selected and investigated further.

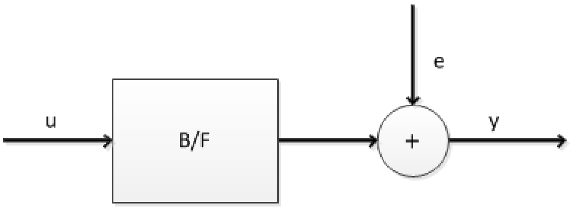


Figure 3.8: Graphical representation of the polynomials in the Output Error model. The mathematical expression can be seen in (3.1).

3.3.1 Output Error

In the Output Error (OE) model structure there were two polynomials, $B(q)$ and $F(q)$ that should be estimated, see Figure 3.8. The OE-estimation function used in Matlab is described in [16].

$$y(t) = \frac{B(q)}{F(q)}u(t) + e(t) \tag{3.1}$$

For the Output Error model structure the following polynomials were estimated:

$$B(z) = 0.001877z^{-9} - 0.001871z^{-10} \tag{3.2}$$

$$F(z) = 1 - 2.791z^{-1} + 1.604z^{-2} + 1.977z^{-3} - 2.6z^{-4} + 0.8099z^{-5} \tag{3.3}$$

When this model was compared against validation data the fit for the 15 experiments were between 91.98-94.67 % which is a good fit. In Figure 3.9 a comparison between simulated output from the model and the validation data can be seen. In Figure 3.10 the poles and zeros can be seen.

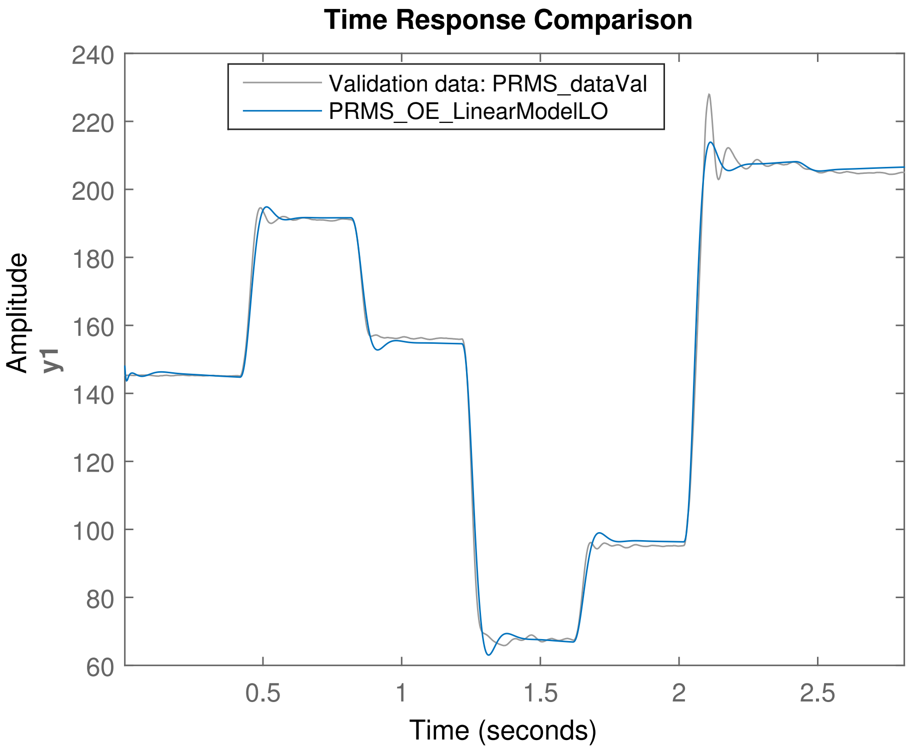


Figure 3.9: Output error model. Fit to validation data: 94.67 %.

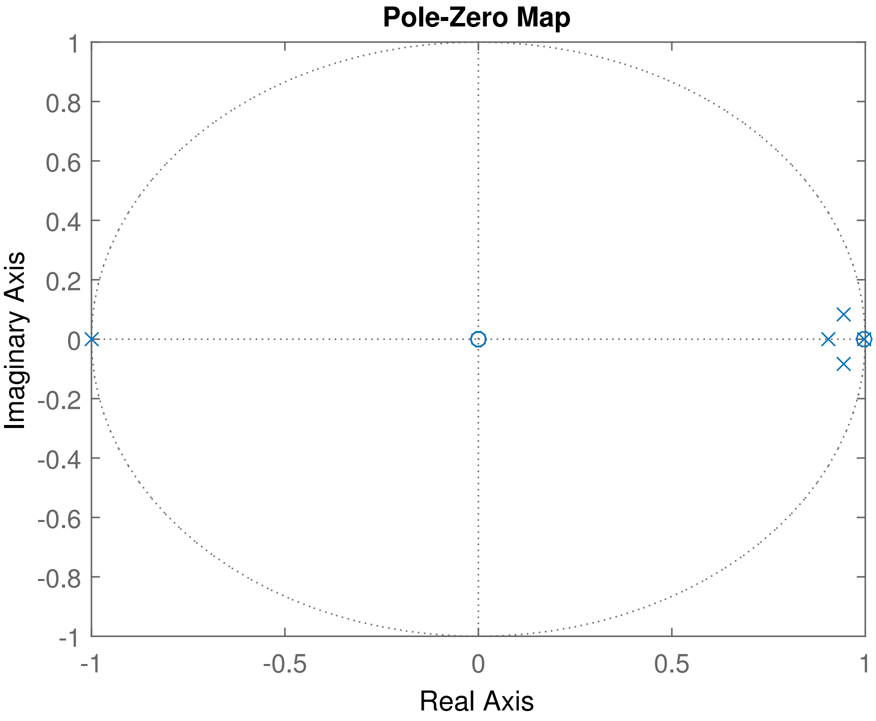


Figure 3.10: Zeroes and poles of the estimated OE-model with the polynomials in (3.2) and (3.3). Poles are marked with x and zeros with o.

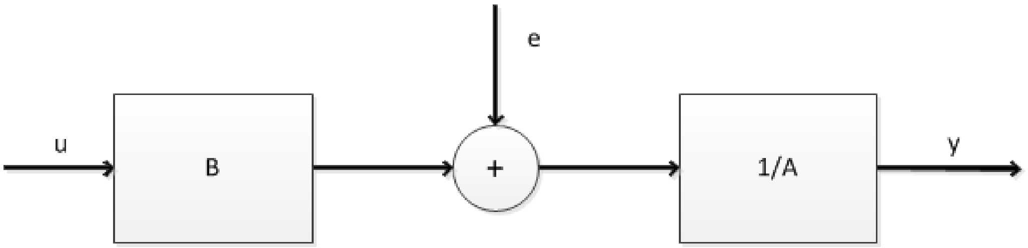


Figure 3.11: The ARX model structure used the polynomials A, B, see (3.4). The signal e is white noise.

3.3.2 ARX

In the ARX model structure there were two polynomials that should be estimated, see (3.4). The ARX estimation function in MATLAB is described in [14]. Figure 3.11 show the graphical expression of the ARX model structure.

$$A(q)y(t) = B(q)u(t) + e(t) \tag{3.4}$$

The same approach as the OE-model was used to find the best permutation of the order of the polynomials. The polynomials estimated can be seen in (3.5) and (3.6).

$$A(z) = 1 - 2.79z^{-1} + 2.931z^{-2} - 1.65z^{-3} + 0.7032z^{-4} - 0.1931z^{-5} \tag{3.5}$$

$$B(z) = 0.001489z^{-9} + 0.0001799z^{-10} - 0.0001148z^{-11} + 3.276e - 05z^{-12} \tag{3.6}$$

In Figure 3.12 a comparison between the validation data and the simulated output of the model can be seen. The placement of the poles and zeroes can be seen in Figure 3.13.

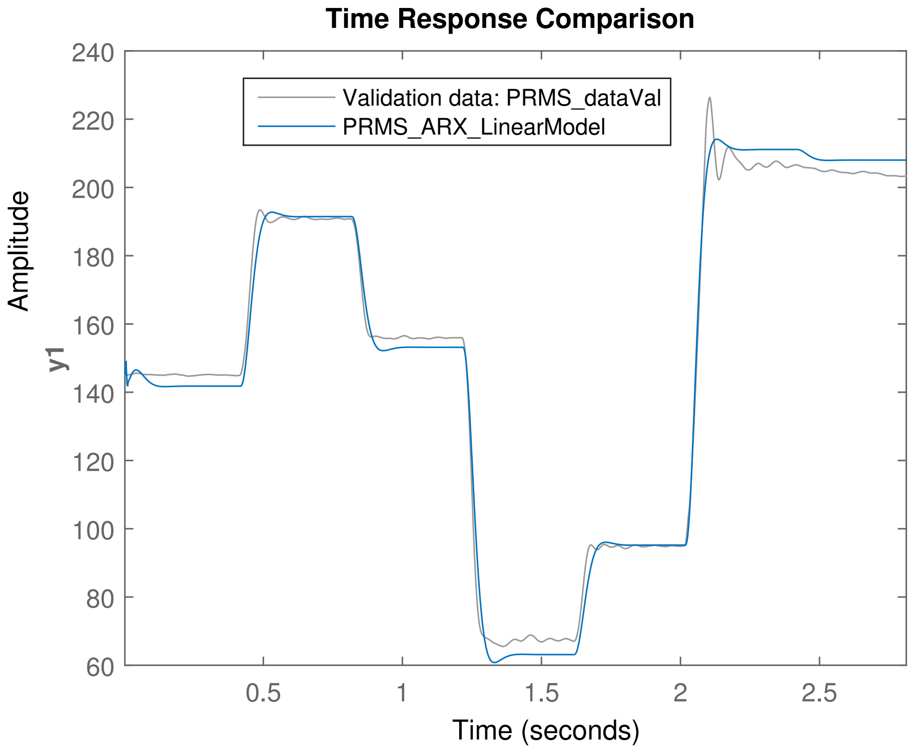


Figure 3.12: ARX model. Fit to validation data: 91.55 %

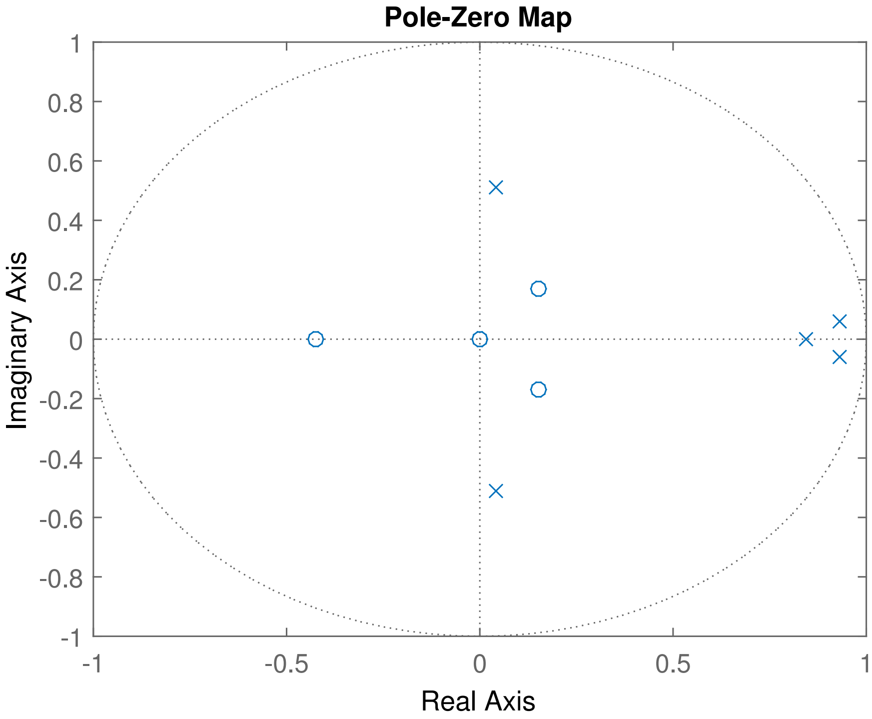


Figure 3.13: Zeroes and poles of the estimated model ARX-model with the polynomials in (3.5) and (3.6). Poles are marked with x and zeros with o.

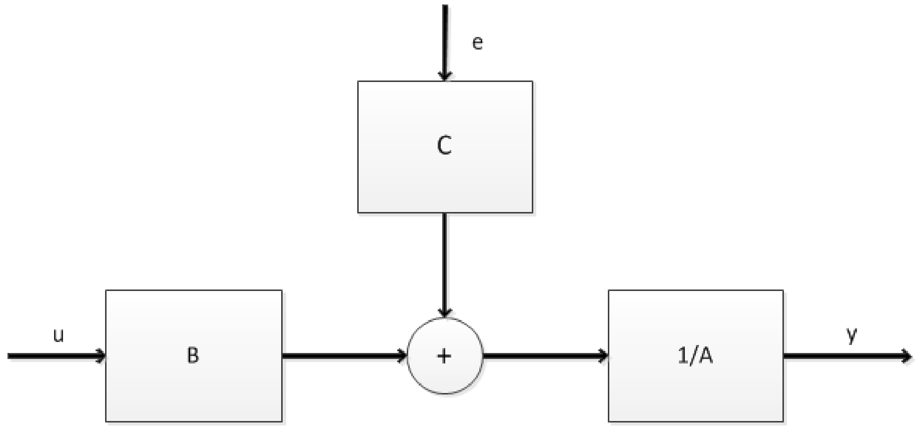


Figure 3.14: The ARMAX model structure used the polynomials A, B and C, see (3.7). The signal e is white noise.

3.3.3 ARMAX

The last linear model structure tested was ARMAX. The difference from the ARX is the extra polynomial at the disturbance signal $e(t)$ which can be seen in (3.7). Figure 3.14 show the graphical expression of the ARMAX model structure where the polynomials A, B and C are estimated. The estimation function in MATLAB for the ARMAX model structure is described in [15].

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{3.7}$$

Again, the best permutation of polynomial orders where calculated and the following polynomials were obtained:

$$A(z) = 1 - 3.831z^{-1} + 5.512z^{-2} - 3.53z^{-3} + 0.8491z^{-4} \tag{3.8}$$

$$B(z) = 0.0004256z^{-9} - 2.064e - 06z^{-10} - 0.0004226z^{-11} \tag{3.9}$$

$$C(z) = 1 - 0.1932z^{-1} - 1.693z^{-2} + 0.1922z^{-3} + 0.6949z^{-4} \tag{3.10}$$

In Figure 3.15 a comparison between the validation data and the simulated output of the model can be seen. The placement of the poles and zeroes can be seen in Figure 3.16.

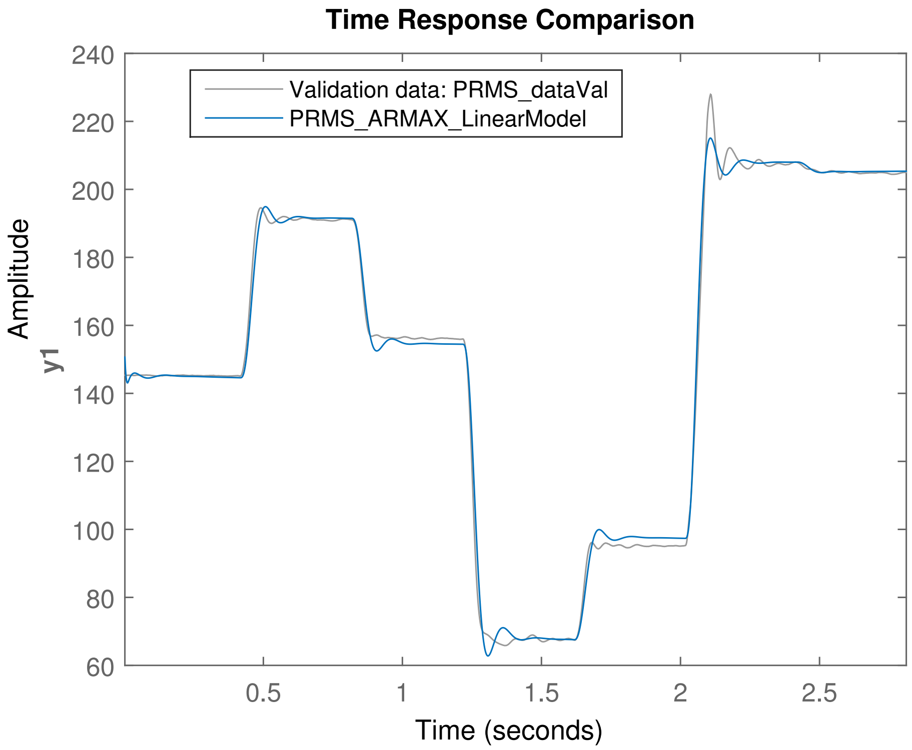


Figure 3.15: ARMAX model fit to validation data: 94.48 %

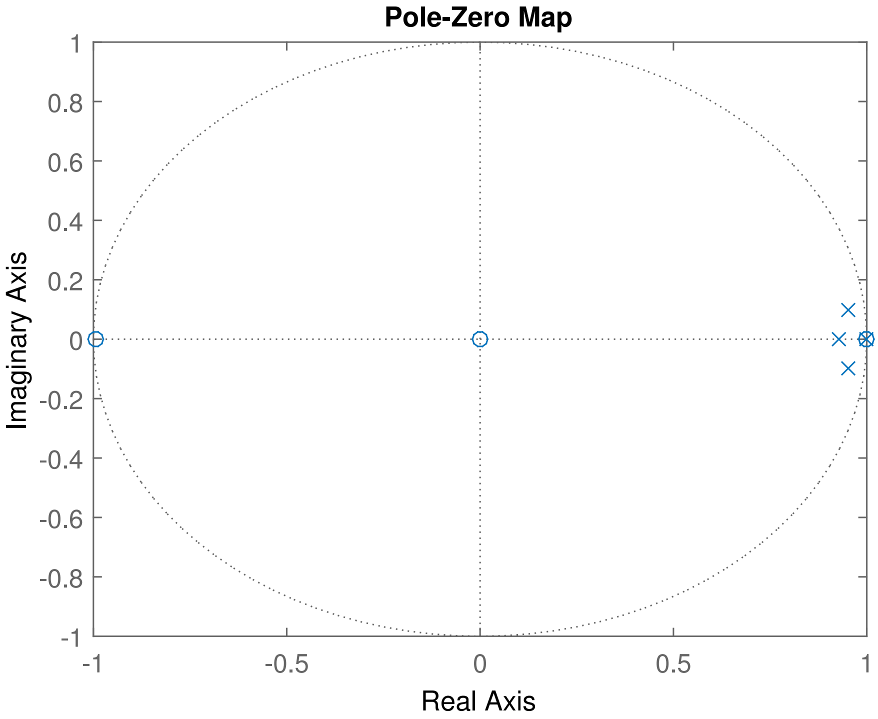


Figure 3.16: Zeroes and poles of the estimated model ARMAX-model with the polynomials in (3.8), (3.9) and (3.10). Poles is marked with x and zeros with o.

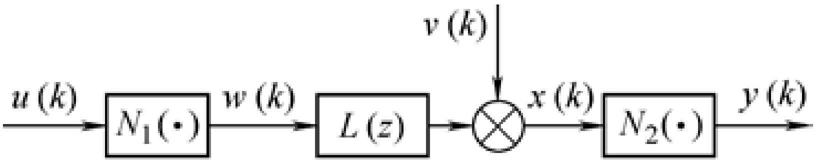


Figure 3.17: Hammerstein-Wiener model. N_1 and N_2 are static nonlinear blocks. L(z) is a linear model of the system.

3.3.4 Hammerstein-Wiener

The last model structure investigated was the nonlinear Hammerstein-Wiener model. The Hammerstein-Wiener model structure can be seen in Figure 3.17. This model structure has one static nonlinearity at the input signal and one static nonlinearity at the output signal. Between these static nonlinearity there are a linear block where the dynamics of the system is captured. In System Identification Toolbox there is a number of static nonlinearities to choose between. All available nonlinearities can be seen in Table 3.1.

1. Piecewise Linear
2. Sigmoid Network
3. Saturation
4. Dead Zone
5. Wavelet Network
6. One dimensional polynomial
7. Custom Network
8. None

Table 3.1: Nonlinearities that can be used by the Hammerstein-Wiener function in Matlab [13].

All of the nonlinearities in Table 3.1 has been tested and the one performing best for this application was the piecewise linear both as input and output nonlinearity. The linear block of the Hammerstein-Wiener model structure only supports OE-models so the model described in Section 3.3.1 was reused to capture the linear dynamics. For the static nonlinearities there were a parameter to set which was called *NumberOfUnits* which specifies the number of breakpoints of the static nonlinearity. To find out the best value for this parameter a script testing all possible combinations between 1-20 were executed. *Ninput*=8 and *Noutput*=15 gave the best result and were used as parameters for the static nonlinearities. The simulated output of the

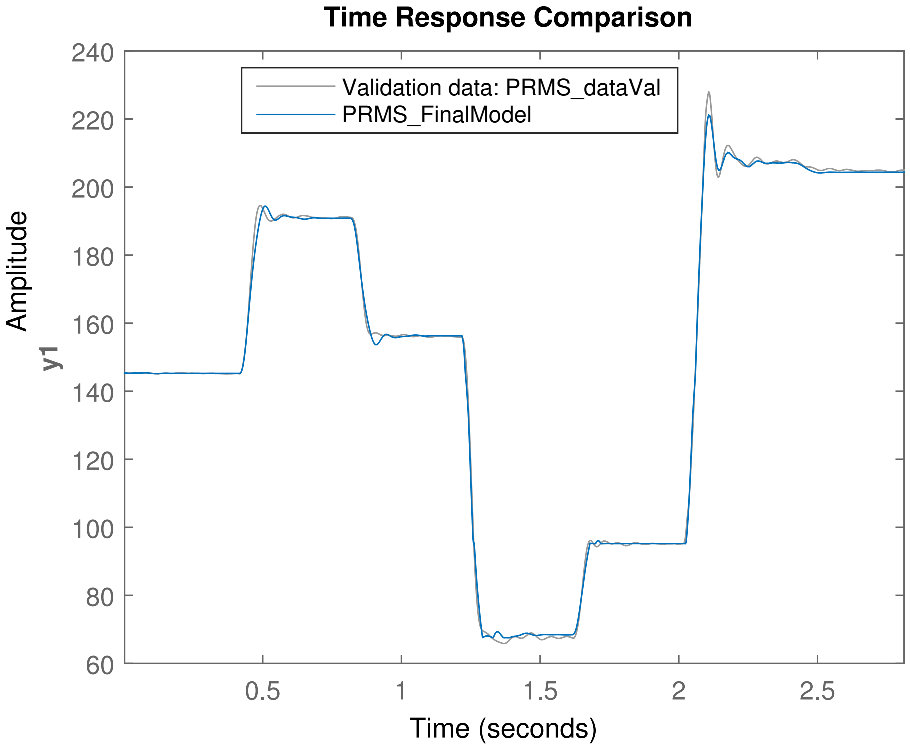


Figure 3.18: Hammerstein-Wiener model - fit to validation data 97.12 %.

Hammerstein-Wiener model compared with validation data can be seen in Figure 3.18.

The estimated Hammerstein-Wiener performs best of all model structures tested when it come to fit against validation data. However, to be able to use this nonlinear model for control in *Control System Toolbox* it need to be linearized around a specific working point according to [10]. Attempts has been done to linearize the Hammerstein-Wiener model and the best linearization do not performs nearly as good as the best estimated linear model. Therefore, the nonlinear Hammerstein-Wiener model will not be used for control.

3.3.5 Summary of Model Estimation

Of the three linear model structures investigated the ARMAX model performed best and was the model structure used to model the plant. The ARMAX model presented in Section 3.3.3 is stable, casual, minimum phase and controllable. The controllability was investigated by calculating the controllability matrix of the state space representation of the ARMAX model. The controllability matrix proved to have full rank which proves that the plant model is controllable.

4

Control Structure and Implementation in Simulink

This chapter presents the implementation of the control structure used for the blank holding pressure of the hydraulic press. The control structure was built in Simulink to be able to create the numerical tuning functionality. The goal was to create a system that imitated the control structure used in the real machine and when this was obtained provide a better set of control parameters.

Closed-loop data from the real machine was collected to have a data to compare with. If the same control parameters were used in simulation as in the real machine and they both behaved the same the control structure was considered correctly implemented. To present the control structure of the blank holding pressure the implementation in Simulink will be explained in the sections below.

4.1 Feedback Control

The feedback control structure of the implementation can be seen in Figure 4.1. The input signal to the system is normally a pseudo random multilevel signal varying between 30-220 bar. The *Controller* block, described in Section 4.3, contains the control logic, feedforward and the PID-controller.

To get the feedback value the pressure need to be measured and filtered. In the real machine there is a delay from the measurement is taken until it appears as the actual value into the controller due to the bus cycle time. Experiments has been done using a oscilloscope to measure the time from the pressure is changed until the controller notice this change. With a scan cycle time of 2 ms the longest time was 4.2 ms and the shortest time 2.3 ms, during a series of 20 measurements. The reason for this delay is the bus cycle time of the Profinet system. This delay was implemented using a simple delay of 1 sample. The sample time of the simulation was set to a fixed step size of 2 ms just like the scan cycle time in the real machine.

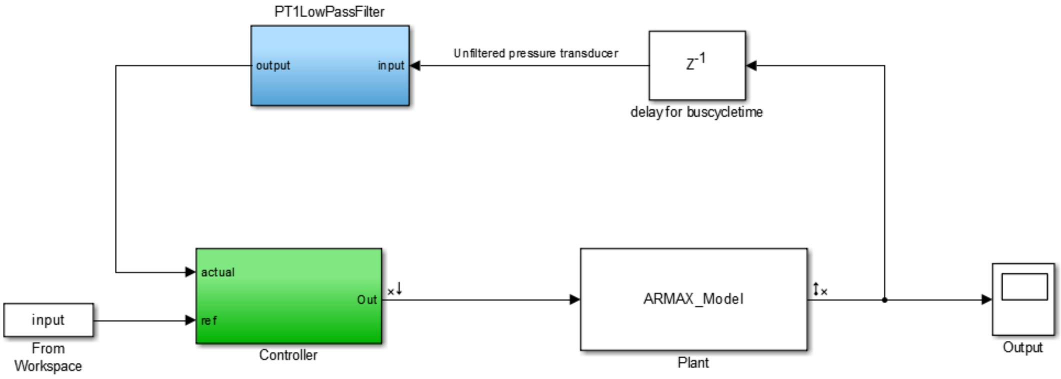


Figure 4.1: Implementation of the control structure for the blank holding pressure. The input and output are in bar.

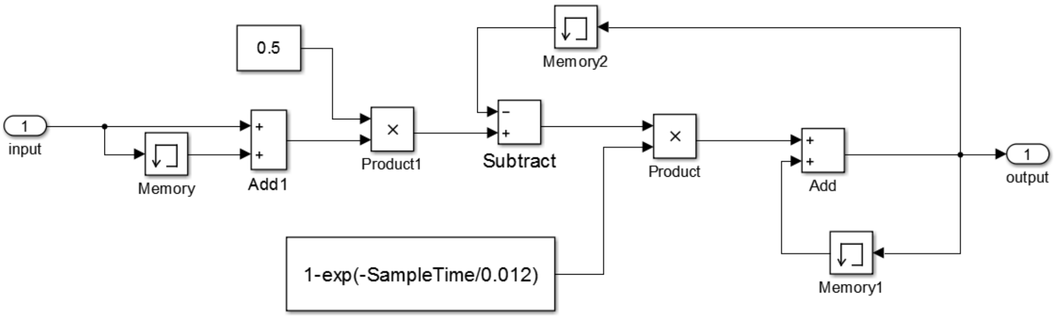


Figure 4.2: Implementation of low pass filter

4.2 Low Pass Filter

A low pass filter was used to filter out high frequency noise before the measurements from the pressure transducer was sent to the controller. In (2.1) the low pass filter is described mathematically. The filter was implemented using this equation and can be seen in Figure 4.2. In Figure 4.3 the filtered and unfiltered signal can be seen.

4.3 Controller

The general control structure used for pressure control in the press can be seen in Figure 4.4. The control structure was implemented using a feedforward PID-controller and can be seen in Figure 4.5. There is a possibility to switch of the PID-controller and only using the feedforward value. In the pressure control of the blank holding pressure the controller is only active when the ratio between actual value and set value is between 90-110 %. Since hydraulic systems are quite slow the PID-controller would be fed with a large error for a long time. This would result in a large control signal which would give a huge overshoot. By activating the controller

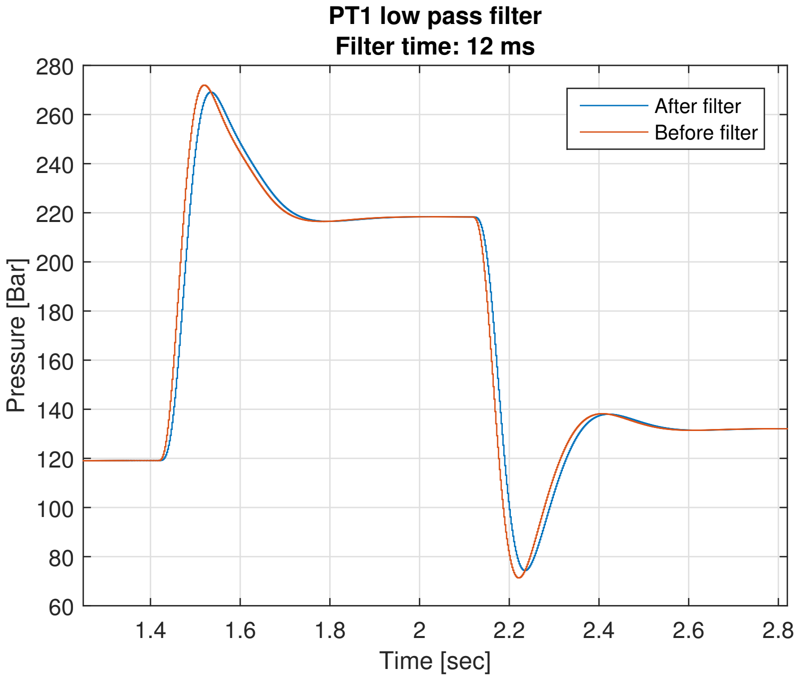


Figure 4.3: Filtered and unfiltered measurement from the pressure transducer.

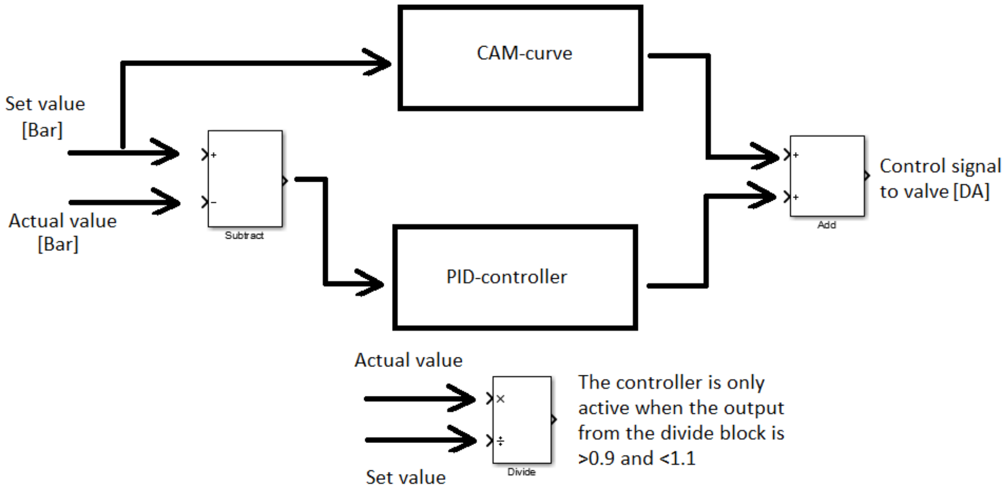


Figure 4.4: General sketch of the control structure

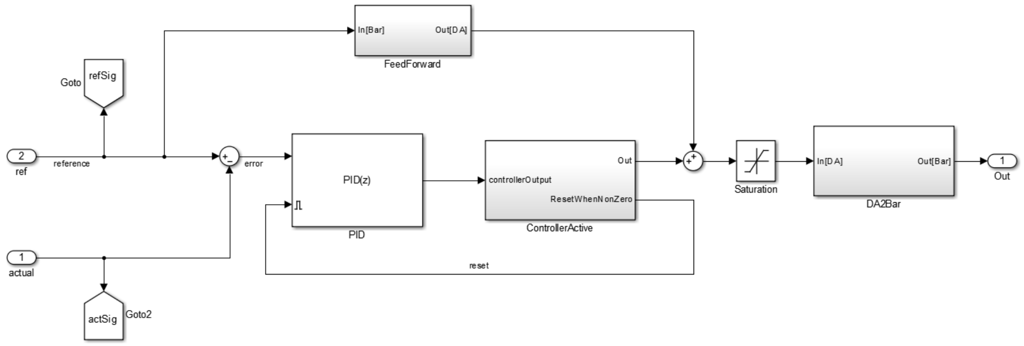


Figure 4.5: The controller subsystem. The PID-controller is discrete with a sample time of 2 ms.

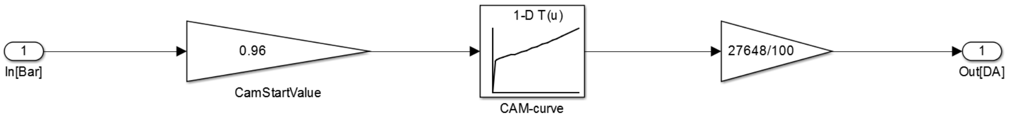


Figure 4.6: The feedforward block. The look-up table is the CAM-curve and the factor 27648/100 convert percentage of volts (from CAM-cure) to DA.

when 90 % is reached this counteracts the huge overshoot. During the time the controller is turned off the feedforward path provide the valve with the set value.

The controller in the real machine is fed with a pressure error and the output from the controller is a DA value that should be added to the feedforward DA value. Since the valve input signal is between 0-10 volt and the largest number DA value the PLC can write to the valve is 27648, this means that 10 volt to the valve is a DA value from the PLC of 27648. To ensure no larger value than 10 volt is written to the valve a saturation of 27648 is added after the feedforward and controller values are added.

In the *ControllerActive* subsystem a logic was built to control the reset of the controller when it is not active. A detailed description of this block can be found in Section 4.3.2.

4.3.1 Feedforward of CAM-value

The subsystem *feedforward* is fed by the set pressure. To convert the value from bar to voltage the CAM-curve is used. In the feedforward path there is a scaling factor called *CamStartValue*, this value can be set by the start-up operator to scale down the feedforward term. The reason for using this scaling constant according to start-up operator D. Magnusson, 5 Feb 2015 is to lower the overshot if the value is set to for example 0.95 instead of 1. In Figure 4.6 the implementation of the feedforward can be seen.

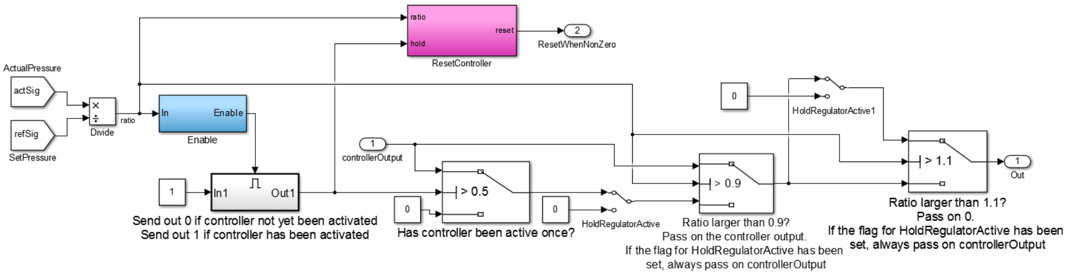


Figure 4.7: Implementation of controller active.

4.3.2 Controller Active

To be able to turn off the controller a logic have been built to reset the controller and also set the output from the controller to zero. In Figure 4.7 this logic can be found. Generally the controller is disabled when the ratio between the actual and set pressure is lower than 0.9 (e.g. when a positive step is added to the set pressure). The controller is also disabled when the ratio is larger than 1.1 (e.g. when a negative step is added to the set pressure). This type of control strategy are used in all pressure controllers in the hydraulic press, e.g the die cushion and the third cylinder. However, there is a flag in the PLC that can be set by the programmer to hold the regulator active after the controller has been activated once. This flag is called *HoldRegulatorActive* and when searching in the PLC code for the pressure control for the blank holding pressure this flag was set. According to programmer P. Sundqvist 28 April 2015 the reason for setting this flag is because the pressure normally do not change more than once during one stroke with the slide. P. Sundqvist also added if the pressure should be changed more than once the flag should not be set.

The level at when the controller should be activated is possible to change by the start-up operator. By default this value is set to 90 % and the start-up operator do not normally change this value.

4.3.2.1 Enable subsystem

The enable block is used to set the flag *HoldRegulatorActive*. When the ratio of 0.9 is hit the block sends an enable signal to the next block. The system identification model has a initial condition of zero. Since the set pressure normally do not start at zero the model has a short settling time. To avoid to set the flag at this time an AND operator has been added to ensure that the *HoldRegulatorActive* flag only can be set after 0.5 seconds which is more than the settling time of the model. In Figure 4.9 the settling time for the model can be seen. The implementation of the enable block can be seen in Figure 4.8.

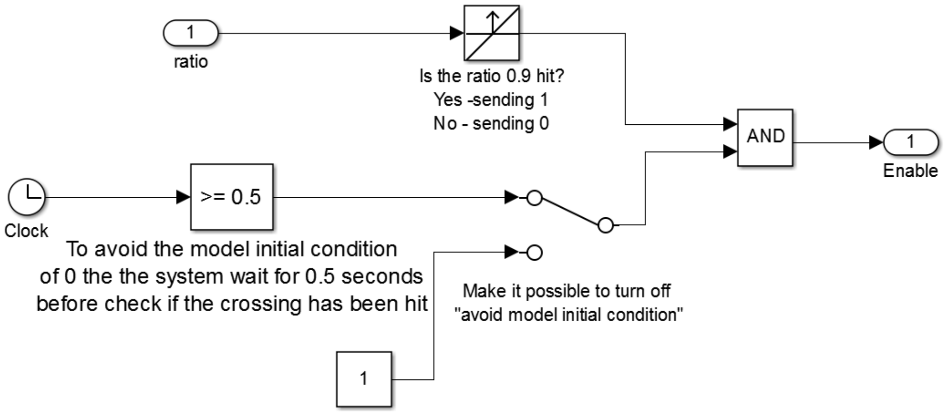


Figure 4.8: Implementation of the enable block. The HitCrossing block is set to 0.9

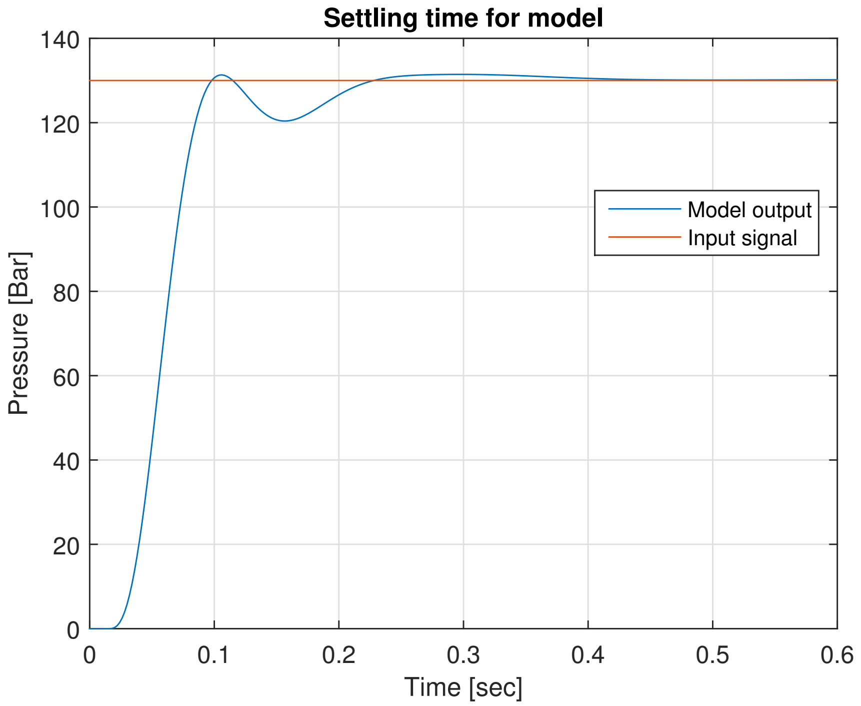


Figure 4.9: The model start from zero but quickly reach up to the input signal.

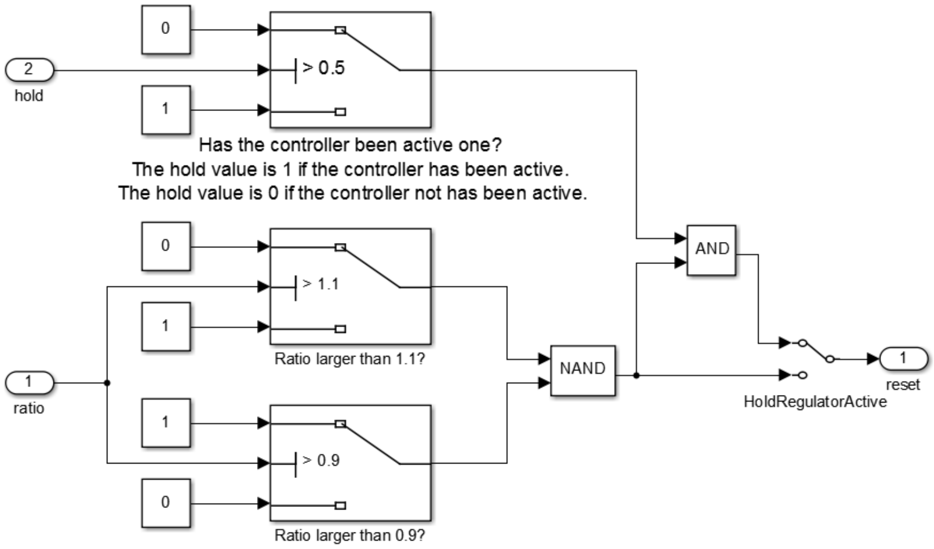


Figure 4.10: Implementation of reset controller. The controller is reset when the value out from the subsystem is nonzero.

4.3.2.2 Reset controller

In the PID-controller block seen in Figure 4.5 there is an input port to reset the controller. When the reset signal is nonzero the controller reset the integral and derivative action to its initial value which is zero. The logic to reset the controller can be seen in Figure 4.10.

4.4 Graphical User Interface

A graphical user interface (GUI) has been created using MATLAB GUI [17] to make it easier to run the simulation and change the setup for every run. A picture of the GUI can be seen in Figure 4.11(b). When the user for example change the *Hold regulator active* value, all switches considering this will change in the simulation logic. The GUI make it possible to quickly set the parameters that should be simulated. There is also a possibility to set and get the PID-controller block parameters P, I & D.

One important notice is the difference between the PID-controller in Simulink and the PID-controller used in the Siemens PLC:

- **P** - no difference between Simulink and Siemens.
- **I** - The value is in seconds in Simulink and milliseconds in the PLC. The I-gain is also inverted in the Siemens world, the lower I-value the larger effect on the control signal. In Simulink the lower I-value the lower impact on the control

signal. This has been solved by inverting the number set to the controller and also to divide the value from Simulink with 1000 to get it in the same unit (ms).

- **D** - The value is in seconds in Simulink but milliseconds in the PLC. The value from Simulink has been divided with 1000 to keep the same unit (ms).

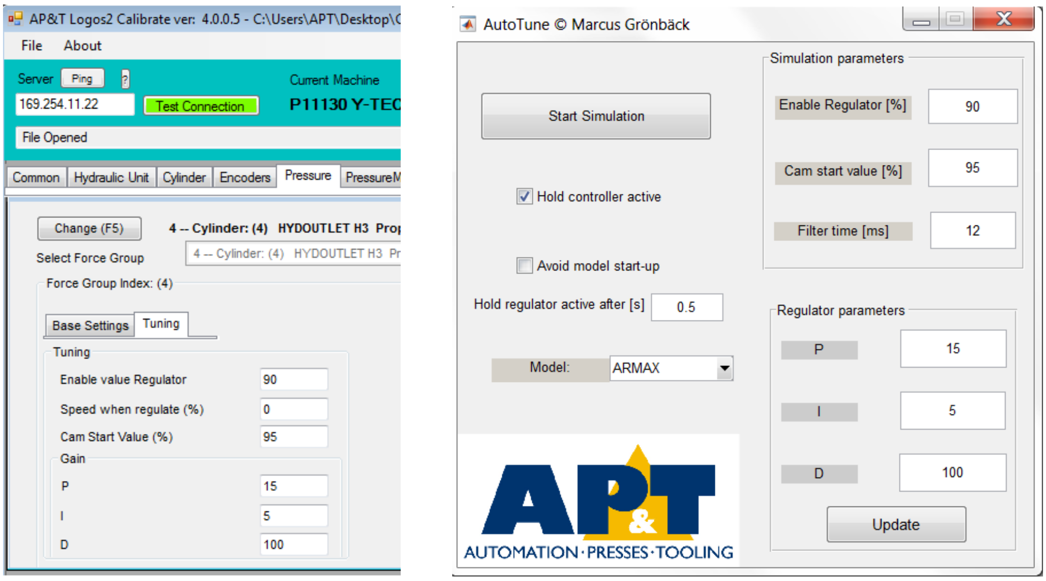
All the modifications of P, I & D values has been programmed into the GUI to easily set a value to the controller in Simulink and also decode what the values represent in the Siemens environment. In Table 4.1 an example of the conversion between Simulink and Siemens environment can be seen.

Parameter	**Simulink**	**Siemens**
P	15	15
I	200	5
D	0.1	100

Table 4.1: Example of conversion of PID parameters between the Siemens and Siemens environment.

4.5 Verification

As mentioned in the beginning of the chapter the verification of a correct implementation would be to compare closed loop data from the real machine with simulation data for the created Simulink model. If the simulation model was set to the same parameters as the real hydraulic press and the same input was used, the output from the simulation and real machine would hopefully look similar to each other. During the start-up phase the start-up operator works with a AP&T designed tool called Calib. It is in Calib the start-up operator does the manual tuning. When the collection of closed loop data was made the start-up operator had tuned the controller according to the settings seen in Figure 4.11(a). The GUI was set to the same values which can be seen in Figure 4.11(b). When a simulation was run and plotted against the output from the real machine the result showed that the control structure was correctly implemented. The verification test can be seen in Figure 4.12.



AP&T Logos2 Calibrate ver: 4.0.0.5 - C:\Users\APT\Desktop\C

File About

Server Ping ?

169.254.11.22 Test Connection

Current Machine P11130 Y-TEC

File Opened

Common | Hydraulic Unit | Cylinder | Encoders | Pressure | PressureM

Change (F5) 4 -- Cylinder: (4) HYDOUTLET H3 Pro

Select Force Group 4 -- Cylinder: (4) HYDOUTLET H3 Pr

Force Group Index: (4)

Base Settings | Tuning

Tuning

Enable value Regulator 90

Speed when regulate (%) 0

Cam Start Value (%) 95

Gain

P 15

I 5

D 100

AutoTune © Marcus Grönbäck

Start Simulation

☑ Hold controller active

☐ Avoid model start-up

Hold regulator active after [s] 0.5

Model: ARMAX

Simulation parameters

Enable Regulator [%] 90

Cam start value [%] 95

Filter time [ms] 12

Regulator parameters

P 15

I 5

D 100

Update

(a) Parameters settings in Calib. The filter time of the pressure transducer can´t be seen but is set to 12 ms.

(b) The GUI settings were the same as in Calib

Figure 4.11: Parameter setting for simulation and data collection in the real machine

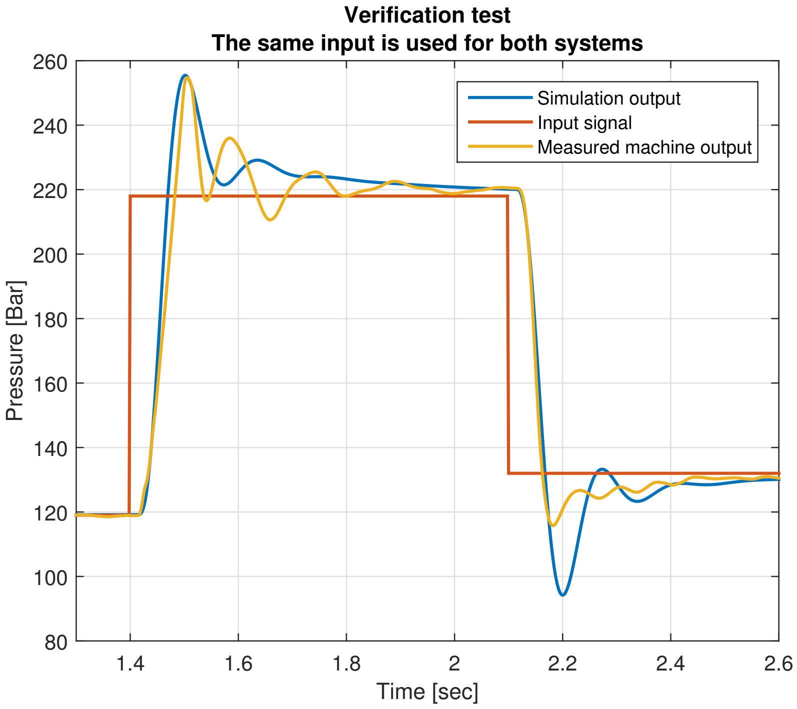


Figure 4.12: The result from the verification test. Both output signals behave similar to each other which indicate a correct implementation of the control structure. The controller is active during the whole time both on the real machine and the simulation.

5

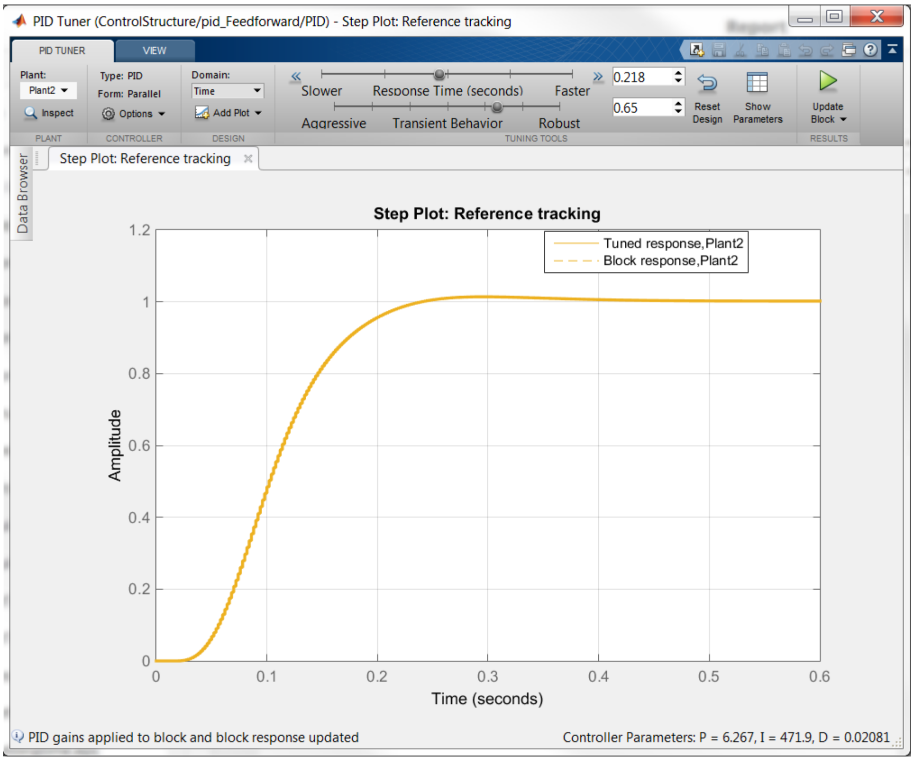
Tuning of Controller

This chapter covers procedure of tuning the controller. The numerical tuning has been performed by the Simulink toolbox *Simulink Control Design*, described in [10]. Simulink Control Design linearize the model and calculate the PID-parameters to achieve a stable system with a fast and robust step response. The closed loop system is designed to have a phase margin of at least 60°. In Figure 5.1 a snapshot from the tuning can be seen. The step is designed to have a small overshot since the feedforward term added after the controller also will create an overshot.

A numerical tuning with the ARMAX plant model presented in Chapter 3 and the control structure described in Chapter 4 has been done and the resulting parameters can be seen in Table 5.1. The simulation result with these parameters can be seen in Figure 5.2. The numerical tuning results in a faster settling time but marginally increase the overshoot compare to the result in Figure 4.12 where the manually tuned PID-parameters are used. Even if the numerical tuning looks promising in simulation it needs to be tested on a real machine to verify if the numerically calculated parameters improves the regulators performance or not compared to the manually tuned. The result from this test can be found in Chapter 6.

Parameter	**Simulink**	**Siemens**
P	6.2674	6.2674
I	471.9176	2.119
D	0.0208086	20.8086

Table 5.1: The tuned PID-parameters.


PID Tuner (ControlStructure/pid_Feedforward/PID) - Step Plot: Reference tracking

Plant: Plant2

Type: PID

Form: Parallel

Domain: Time

Response Time (seconds): 0.218

Transient Behavior: 0.65

PID gains applied to block and block response updated

Controller Parameters: P = 6.267, I = 471.9, D = 0.02081

Figure 5.1: The PID-gains are calculated with the corresponding step response in Simulink Control Design

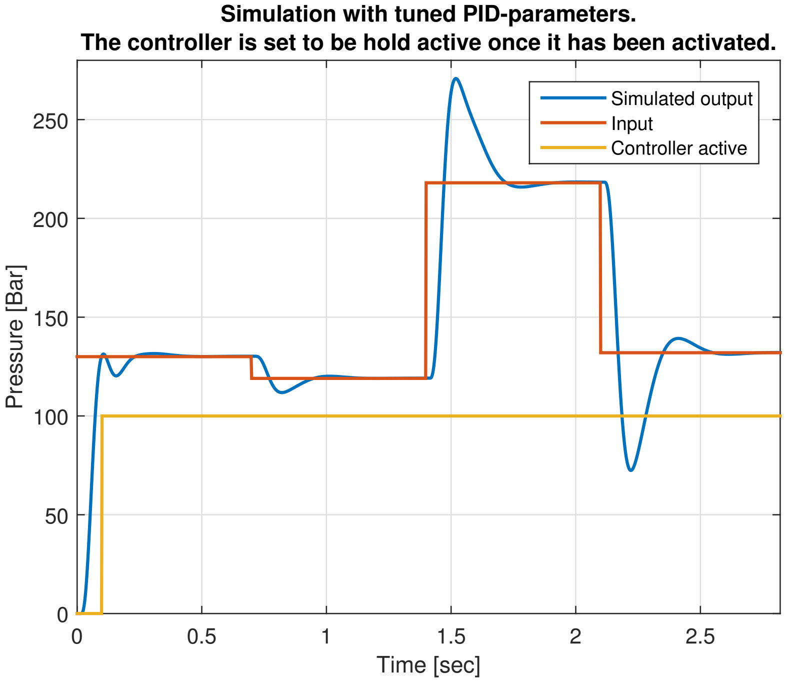


Figure 5.2: Simulation result with the PID-parameters seen in Table 5.1. The controller is hold active once it has been activated.

6

Result

This chapter cover the results obtained in the thesis. To verify if the calculated PID-parameters results in a better performance, numerically tuned parameters were tested on a real machine. Unfortunately, the machine previously used to collect data from has been delivered to the customer. An other machine had to be used for the final test. This means that new data need to be collected and a new model estimated. However, the machine used for the final test was a copy of the previously used machine which means that the system described in Chapter 2 still is valid. The only different found was that the sample time in the PLC was 4 ms instead of 2 ms in the previous machine. The sample time of the Simulink model was set to 4 ms as well as the sample time for the discrete PID-controller. The bus cycle time delay remained 1 sample due the fact that the delays was measured in an oscilloscope to vary between 4.3 and 8.2 ms.

In the final test the following steps were taken:

1. Collect new open-loop data.
2. Create a new model using system identification.
3. Insert the model into the Simulink model described in Chapter 4.
4. Tune the discrete PID-controller.
5. Simulate the system with the tuned parameters.
6. Test the new parameters on the real machine.
7. The start-up operator conclude if the performance has improved or not compared to the manual tuning.

6.1 Data Collection

The data collection was done in the same way described in Section 3.1. The same 300 mm long hydraulic cylinder was used and the slide velocity was set to 90 mm/s.

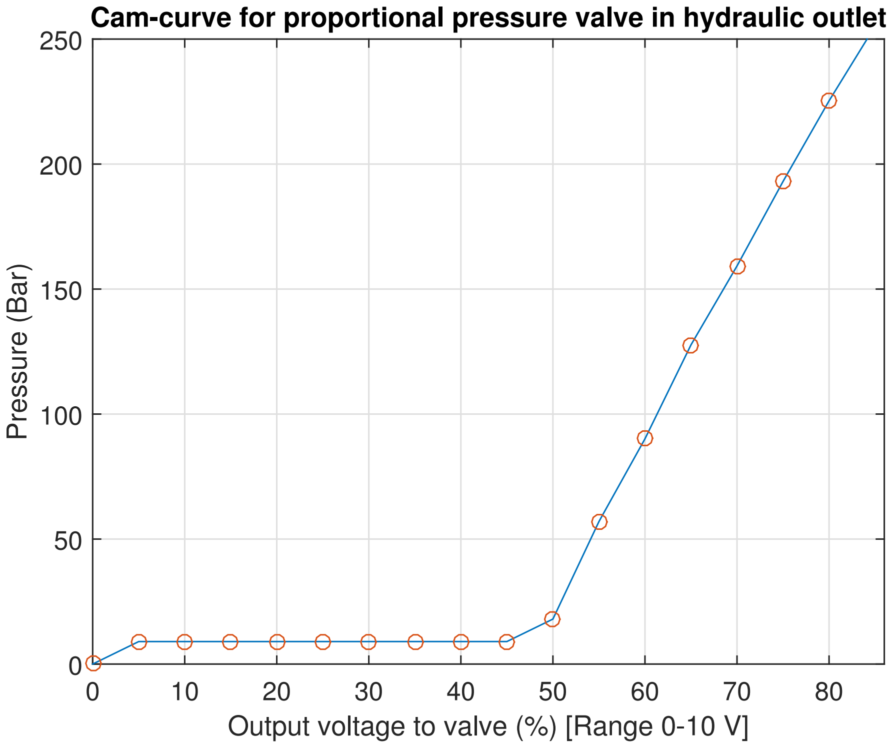


Figure 6.1: The measured CAM-curve for proportional pressure valve of the hydraulic outlet.

The input signals to the set pressure were the same as in previous data collection and can be seen in Figure A.1 and A.2 in Appendix A. The collected data can be seen in Figures C.1 and C.2 in Appendix C. The data was split up into estimation data and validation data in the same way described in Section 3.1.1. 10 experiments were placed in the estimation data file and 10 experiments in the validation data file. The CAM-curve also had to be collected since the CAM-curve differs between valves. The CAM-curve for the proportinal pressure valve used in the final test can be seen in Figure 6.1.

6.2 Create Model

The model was estimated in the same way as described in Section 3.3. The mean and offset of the collected data were kept since this proved to create a better model than if the mean and offset should be removed. The model structure selected was an ARMAX model. The script testing all permutations of order of the polynomials were run and the following polynomials were obtained:

$$\begin{aligned} A(z) = 1 - 4.166z^{-1} + 7.952z^{-2} - 10.48z^{-3} + 11.62z^{-4} \\ -10.72z^{-5} + 7.887z^{-6} - 4.638z^{-7} + 1.932z^{-8} - 0.3886z^{-9} \end{aligned}$$

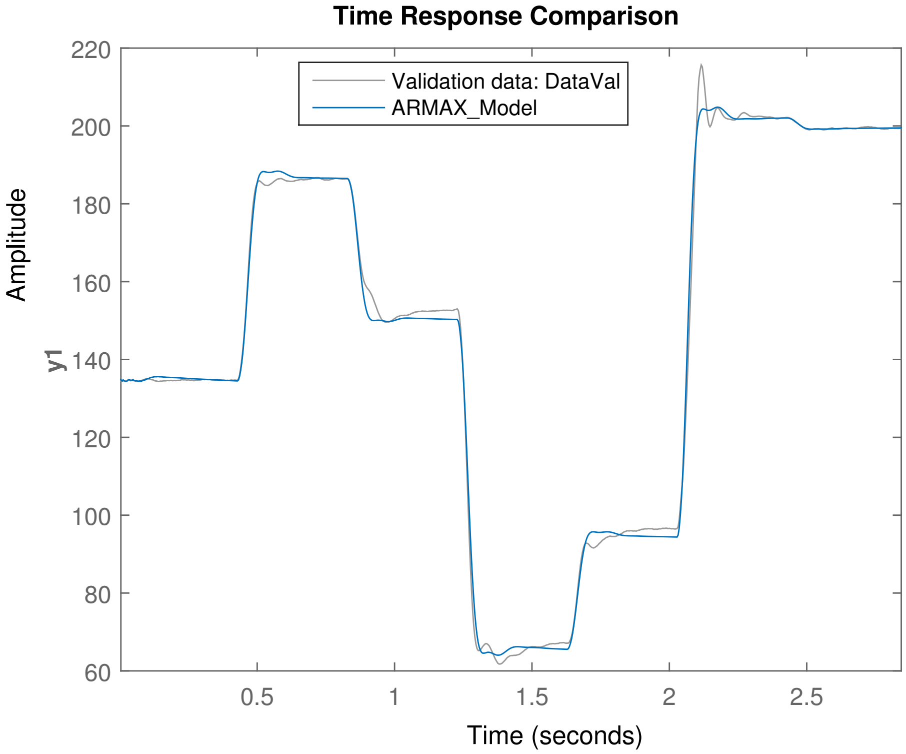


Figure 6.2: Comparison between the created ARMAX model and validation data. The fit to validation data is between 92-95 % for the 10 experiments.

$$B(z) = 0.004842z^{-7} - 0.003177z^{-8} - 0.003713z^{-9} + 0.002061z^{-10}$$

$$C(z) = 1 - 1.021z^{-1} - 0.07624z^{-2} + 0.06594z^{-3} - 0.4009z^{-4} + 0.6291z^{-5} - 0.4958z^{-6} + 0.3007z^{-7}$$

In Figure 6.2 a comparison between the estimated model and validation data can been seen. The poles and zeros of the model can be seen in Figure 6.3. The estimated plant model was stable, causal, minimum phase and controllable just like the ARMAX plant model estimated in Section 3.3.3.

The model was inserted in the Simulink model described in Chapter 4. The PID-controller tuning procedure presented in Chapter 5 was used and the parameters seen in Table 6.1 were obtained.

Parameter	**Simulink**	**Siemens**
P	12.0132	12.0132
I	439.4252	2.2757
D	0.082106	82.106

Table 6.1: The tuned PID-parameters in the final test.

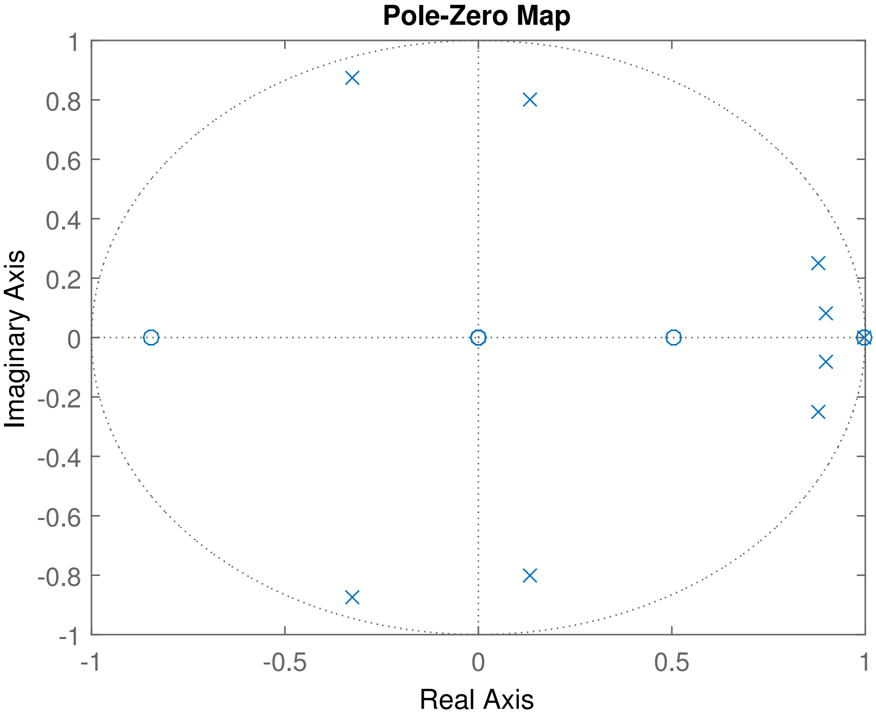


Figure 6.3: Zeroes and poles of the estimated model ARMAX-model. Poles is marked with x and zeros with o.

6.3 Test on Real Machine

The parameters in Table 6.1 were given to a start-up operator to be tested in the same machine and on the same hydraulic outlet as the open-loop data had been collected from. The start-up operator had already manually tuned in the PID-controller for the blank holding pressure. According to start-up operator D Staafjord 12 May 2015 the manual tuning he had done was really good and he was sceptical if a better tuning could be achieved. The test with the numerically tuned PID-parameters versus the manual PID-parameters can be seen in Figures 6.4 and 6.5. When D Staafjord 12 May 2015 saw the result he concluded that the numerically tuned parameters undoubtedly performed better than the manually tuned parameters. D Staafjord was impressed and admitted defeated by the mathematics although he has 12 years experience tuning PID-controllers.

To verify the correctness of the Simulink model the same step response as in the real machine was simulated with the same PID-parameters. These results can be seen in Figures 6.6(a) and 6.6(b).

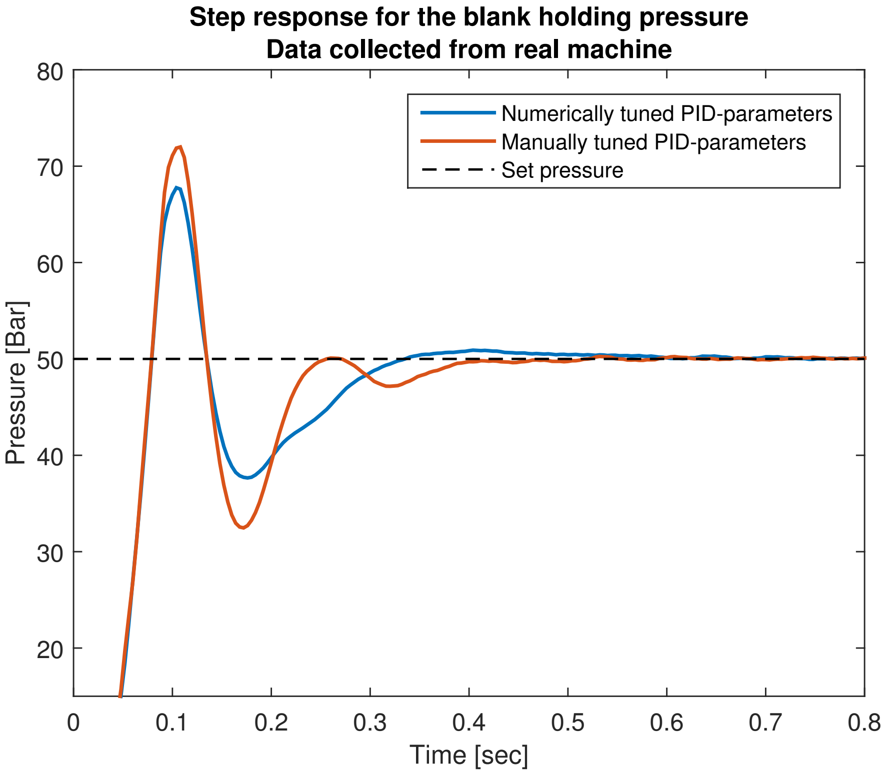


Figure 6.4: The numerically tuned PID-parameters results in a faster settling time and lower overshot. The experiments are done in the same machine with the same oil temperature and same speed on the slide.

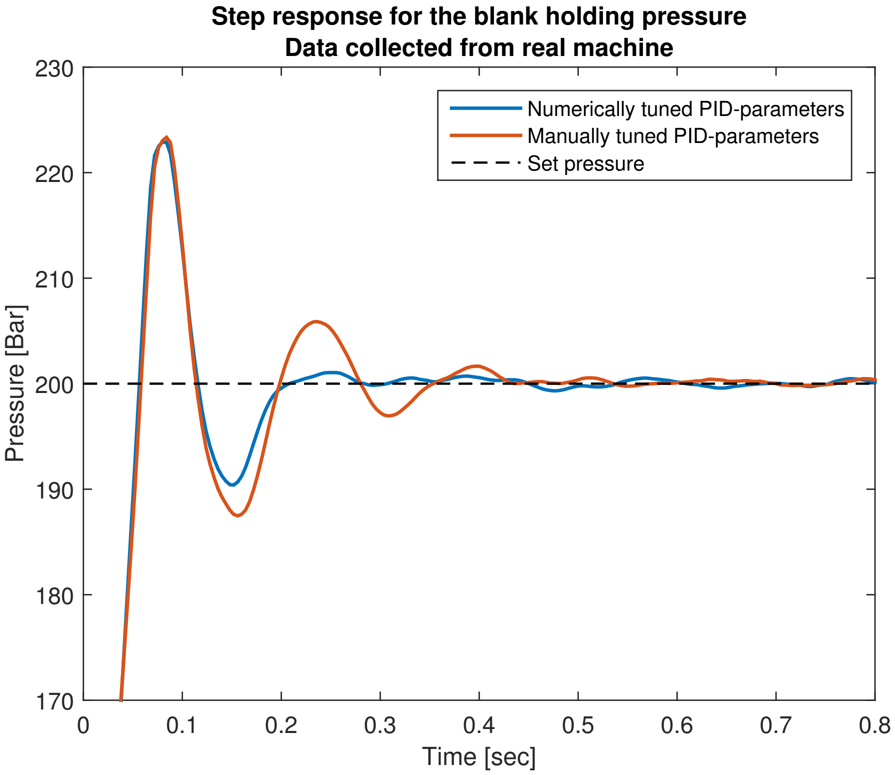
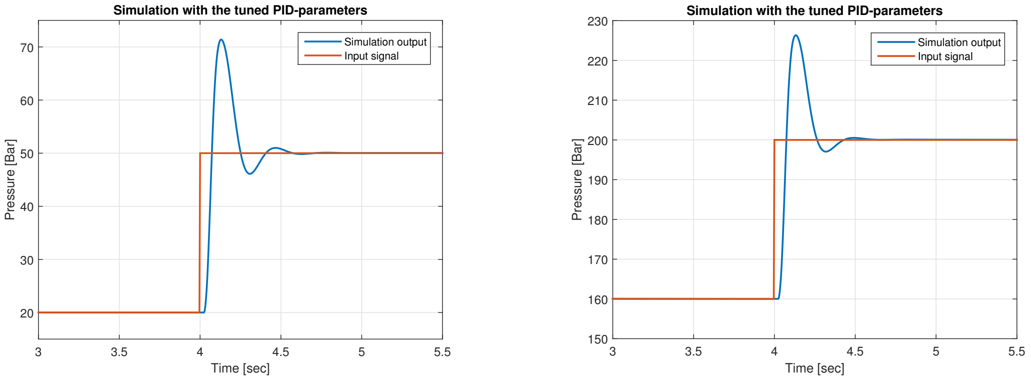


Figure 6.5: With a set pressure of 200 bar the numerically tuned parameters give a faster settling time and lower undershot than the manually tuned parameters.



(a) Same step as Figure 6.4.

(b) Same step as Figure 6.5.

Figure 6.6: Simulation with the numerically tuned parameters seen in Table 6.1. The simulated step responses looks similar to the step responses seen in Figures 6.4 and 6.5.

7

Discussion and Conclusion

In this chapter the obtained results are discussed and future work is presented. The chapter also include the conclusions drawn based on the the results presented in this thesis.

The results obtained are satisfying for all parts involved. An idea of a possible solution of a problem ended up in functioning method of tuning the PID-controllers. In this master thesis the blank holding pressure for a hydraulic outlet has been studied. However, in the future all load sensing hydraulic systems in the press controlled by a PID-regulator can take advantage of the result presented in this thesis. Both the die cushion and third cylinder use the same hydraulic system for the blank holding pressure.

A script has been developed in Matlab where the open-loop data are imported and a model is estimated. This script test a number of permutations of the model and save the model with best fit to validation data. This script might not give the best insight in the system identification process but since this tool is developed for start-up operators this might even be an advantage. The start-up operators do not have any knowledge of system identification and do not want to learn it to use this tool. The tool need to be easy to use and understand. The start-up operator do not want to know the all theory behind the tool, they just need it to work. The start-up operator works in a stressful environment where every minute counts. If the tuning tool should be too complex the start-up operator would tune the controller manually as they use to do and not use the numerical tuning tool.

The control structure for the pressure control for the blank holding pressure has been developed to fit for hydraulic systems. Turning off the controller and only send the CAM-value seems to be a good way of controlling the system. No changes to the control structure has been presented in this thesis but the focus has been to improve the existing system with the numerical tuning procedure.

When the result described in Chapter 6 was produced a different machine had to be used. Since the machine had the same hydraulic design the change of machine had no effect on the result. The only thing different between the machines were the sampling time of the PLC. This change does not affect the result since both the manual tuning and numerical tuning has been done with the same sample time.

Because a new machine had to be used this gave an opportunity to test the whole chain of steps from data collection, model building to numerical tuning.

7.1 Future work

Even if result is promising there is a few things that still remains until the numerical tuning can be fully adapted by the start-up operators.

Today, the manual tuning takes shorter time then the numerical tuning procedure. In particular the data collection of open-loop data takes long time which needs to be reduced. If the open-loop data collection would be implemented in the start-up operators software Calib this process would be much quicker. A vision is to create a button in Calib the start-up operator can hit and a preprogrammed data collection experiment starts and save the input-output data in a .CSV file which can be exported to MATLAB. The development of this functionality has started and will in the near future be a part of Calib. In Section 1.3 the tuning procedure is set to take a maximum of 10 minutes of the start-up operator's time. This requirement can not be considered satisfied since the data collection take to long time. However, the script estimating the model and the tuning of controller is quick so if the data collection is speeded up by the data collection program the requirement of maximum 10 minutes will probably be met.

The recommendation is to create a new model for every numerical tuning. The dynamics might differ even if the design of the hydraulic system is the same. Further work is needed to determine how much the models differ between two hydraulic systems designed in the same way. If a general model can be created this would save a lot of time. However, if the same model would be used the numerical tuning would result in the exact same PID-parameters which rarely is the case according to start-up operator D Magnusson 20 jan. 2015. Every tuning of the blank holding pressure of a hydraulic outlet requires a unique set of parameters. Two different hydraulic presses has been used in this thesis. Both machines had the same hydraulic design but different sampling time which make it impossible to compare the two created models.

To be able to create a model and tune the controller Matlab need to be used. One MATLAB license with the required toolboxes cost approximately 73000 kr which is a quite large investment cost. The manager need to decide if all start-up operators should have a license or the collected data should be passed on to a simulation engineer with a MATLAB license that create the model, tune the controller and pass on the calculated parameters to the start-up operator.

7.2 Conclusion

The results obtained in this thesis states that the presented approach of modeling the system and tuning the controller works. The contribution of the thesis is a proof-of-concept where a controller has been numerically tuned with improved performance compared to manual tuning.

The overall aim with the numerical tuning functionality was to reduce the lead time. This can not be considered reached but a large step has been taken during this thesis work. The manager sees major savings in lead time and quality improvements when the numerical tuning tool is fully implemented in the near future.

Bibliography

[1] L. Ljung *System identification: Theory for the User*, Prentice hall, 1999

[2] Ling, T.G. el al. (2011) System identification of Electro-Hydraulic Actuator Servo System in *International Conference on Mechatronics*; 17-19 May, 2011, Kuala Lumpur, Malaysia

[3] Jelali, M and Kroll, A (2003) *Hydraulic Servo systems - Modelling, Identification and Control*, London: Springer

[4] R. Neugebauer et al. (2012) Presshardening — An innovative and challenging technology in *Archives of civil and mechanical engineering 12* ; April, 2012, Chemnitz, Germany

[5] Kim D. H. and Tsao T-C (1998) Identification and control of Electrohydraulic Actuator Modeled as a linear periodic system in *Proceedings of the American Control Conference*; June, 1998, Philadelphia, PA, USA

[6] Rozali S. et al. (2010) PID Controller Design for an Industrial Hydraulic Actuator with Servo System in *Proceedings of 2010 IEEE Student Conference on Research and Development*; 13 - 14 Dec 2010, Putrajaya, Malaysia

[7] Yan J., et al. (2013) Nonlinear Modeling and Identification of the Electrohydraulic Control System of an Excavator Arm Using BONL Model, *CHINESE JOURNAL OF MECHANICAL ENGINEERING*, Vol. 26, No. 6, 1212-1221

[8] L. Ljung and T. Glad, *Modeling of dynamic systems*, Upper Saddle River NJ: Prentice hall, 1994

[9] Mathworks, "System Identification Toolbox" 2015, [Online]. Avaliable: `http://se.mathworks.com/products/sysid/`, Date of access: 23 Jan. 2015

[10] Mathworks, "Control System Toolbox" 2015, [Online]. Avaliable: `http://se.mathworks.com/products/control/`, Date of access: 23 Jan. 2015

[11] Mathworks, "Handling Offsets and Trends in Data" 2015, [Online]. Avaliable: `http://se.mathworks.com/help/ident/ug/handling-offsets-and-trends-in-data.html`, Date of access: 15 Mar. 2015

[12] Mathworks, "Estimate time delay (dead time) from data" 2015, [Online]. Available: `http://se.mathworks.com/help/ident/ref/delayest.html`, Date of access: 26 Mar. 2015

[13] Mathworks, "Estimate a Hammerstein-Wiener model" 2015, [Online]. Avaliable: `http://se.mathworks.com/help/ident/ref/nlhw.html`, Date of access: 27 Mar. 2015

[14] Mathworks, "Estimate parameters of ARX or AR model using least squares" 2015, [Online]. Avaliable: `http://se.mathworks.com/help/ident/ref/arx.html`, Date of access: 27 Mar. 2015

[15] Mathworks, "Estimate parameters of ARMAX model using time-domain data" 2015, [Online]. Avaliable: `http://se.mathworks.com/help/ident/ref/armax.html`, Date of access: 27 Mar. 2015

[16] Mathworks, "Estimate Output-Error polynomial model using time or frequency domain data" 2015, [Online]. Avaliable: `http://se.mathworks.com/help/ident/ref/oe.html`, Date of access: 27 Mar. 2015

[17] Mathworks, "Creating Apps with Graphical User Interfaces in MATLAB" 2015, [Online]. Avaliable: `http://se.mathworks.com/discovery/matlab-gui.html`, Date of access: 06 May 2015

A

Input signals

Input signals used for the system identification described in Chapter 3.

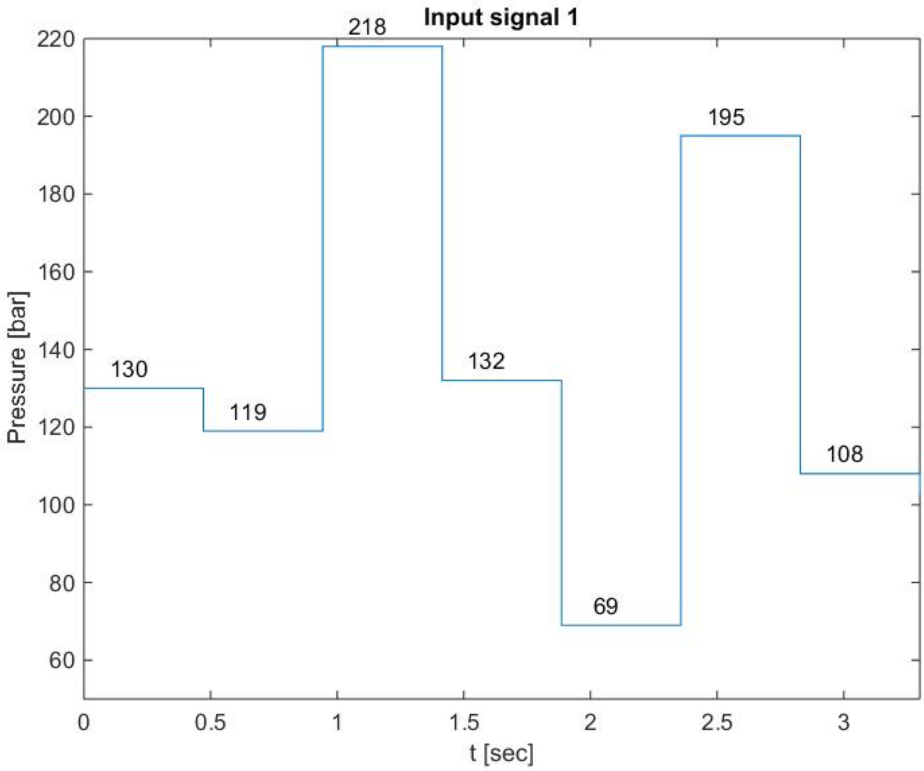


Figure A.1: Input signal 1

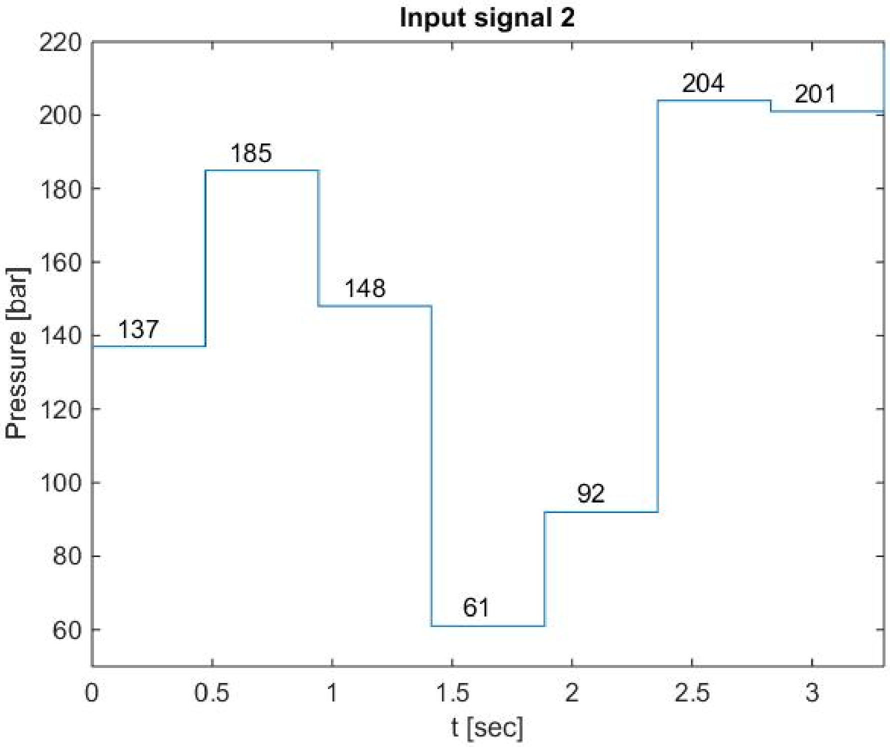


Figure A.2: Input signal 2

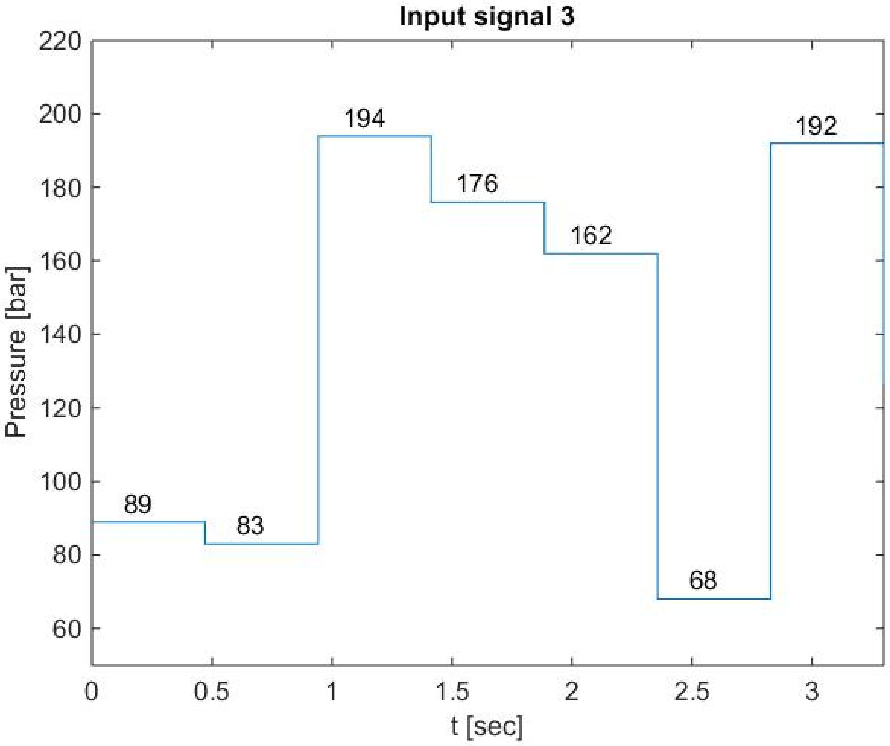


Figure A.3: Input signal 3

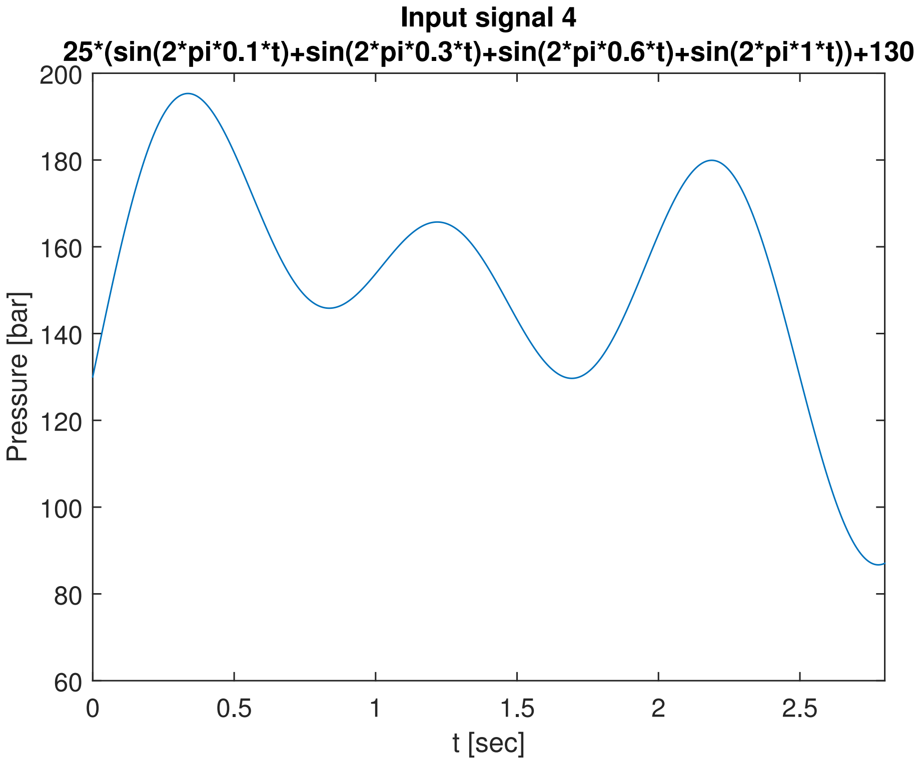


Figure A.4: Input signal 4

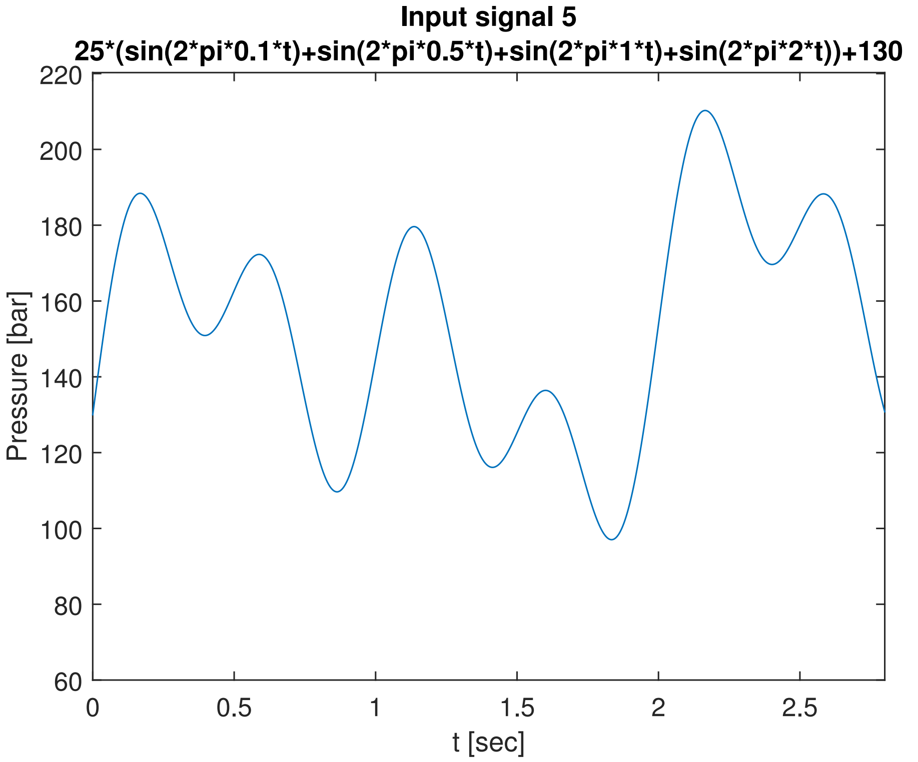


Figure A.5: Input signal 5

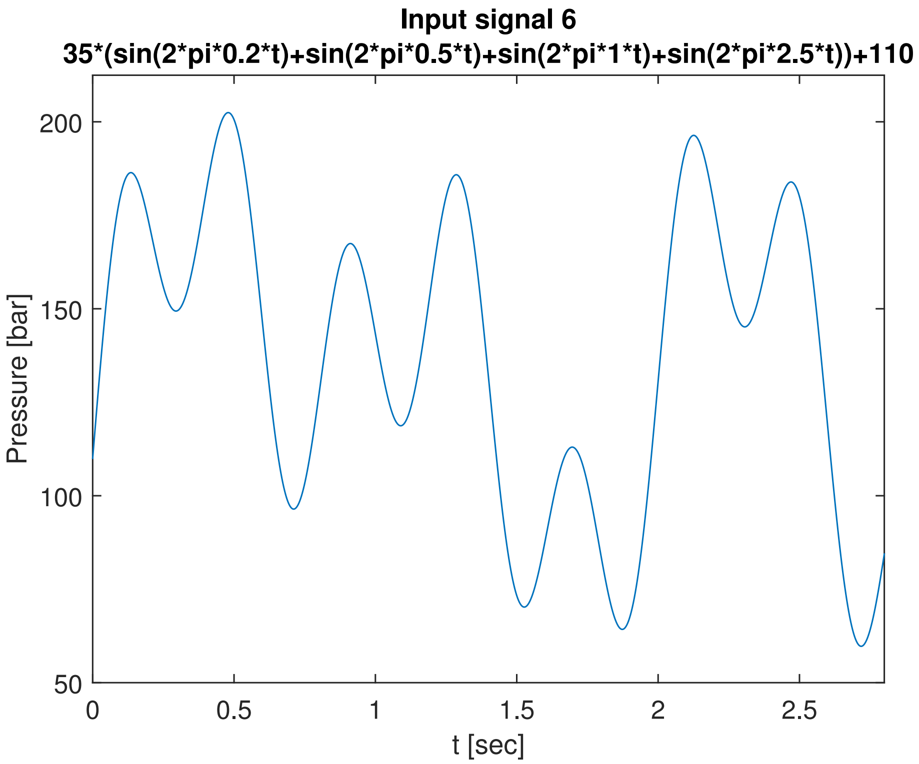


Figure A.6: Input signal 6

B

Input/output data first collection

The following figures show the data collected from the hydraulic press used to build the plant model described in Chapter 3.

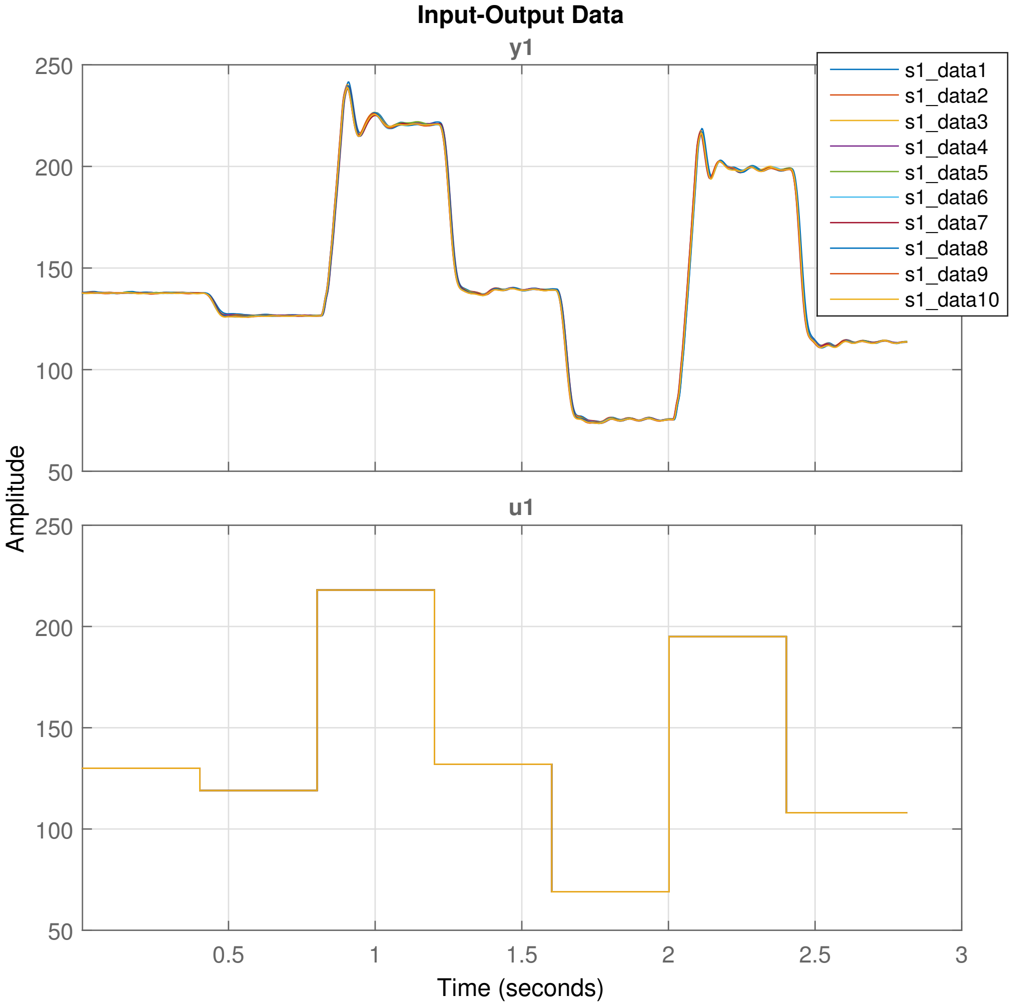


Figure B.1: 10 different strokes with input signal 1.

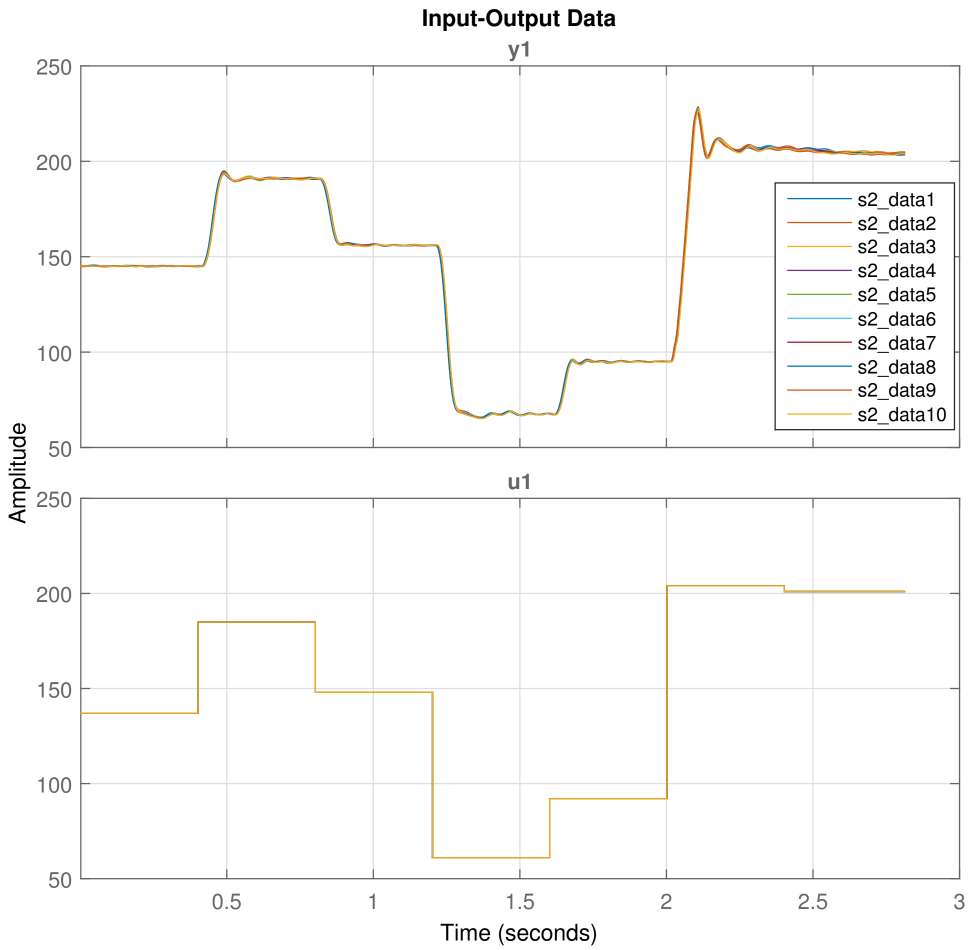


Figure B.2: 10 different strokes with input signal 2.

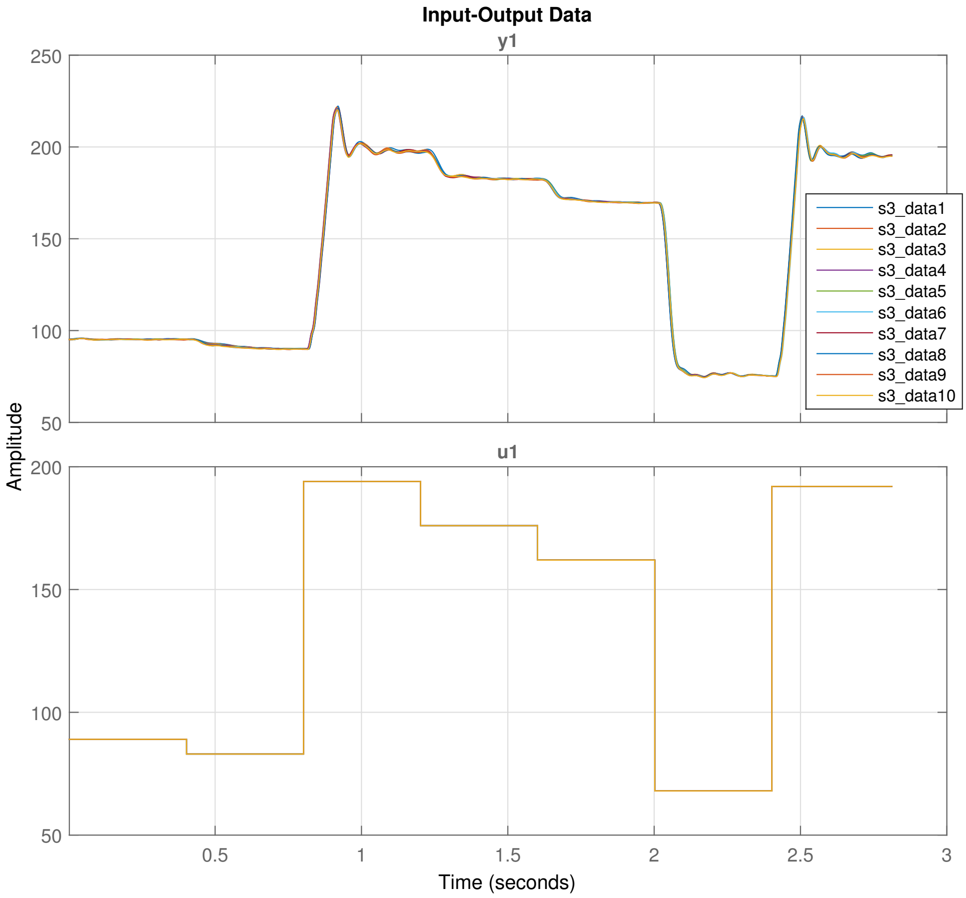


Figure B.3: 10 different strokes with input signal 3.

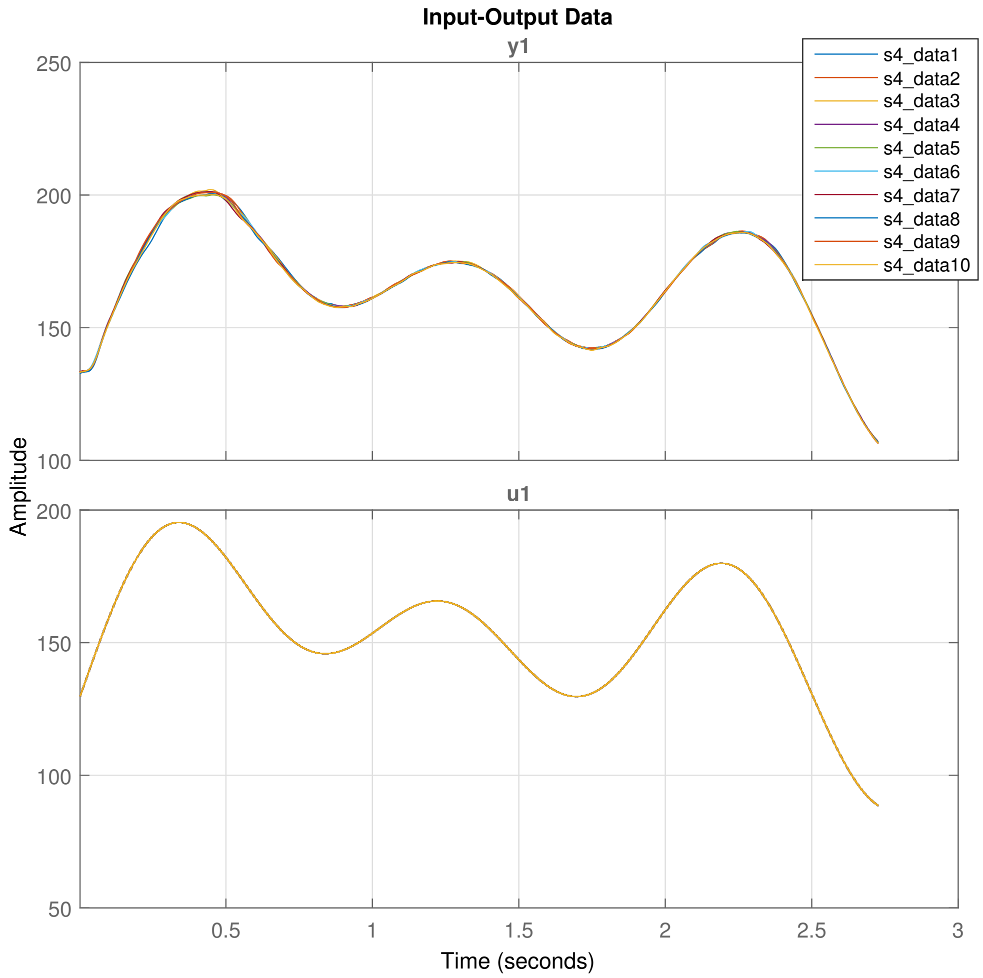


Figure B.4: 10 different strokes with input signal 4.

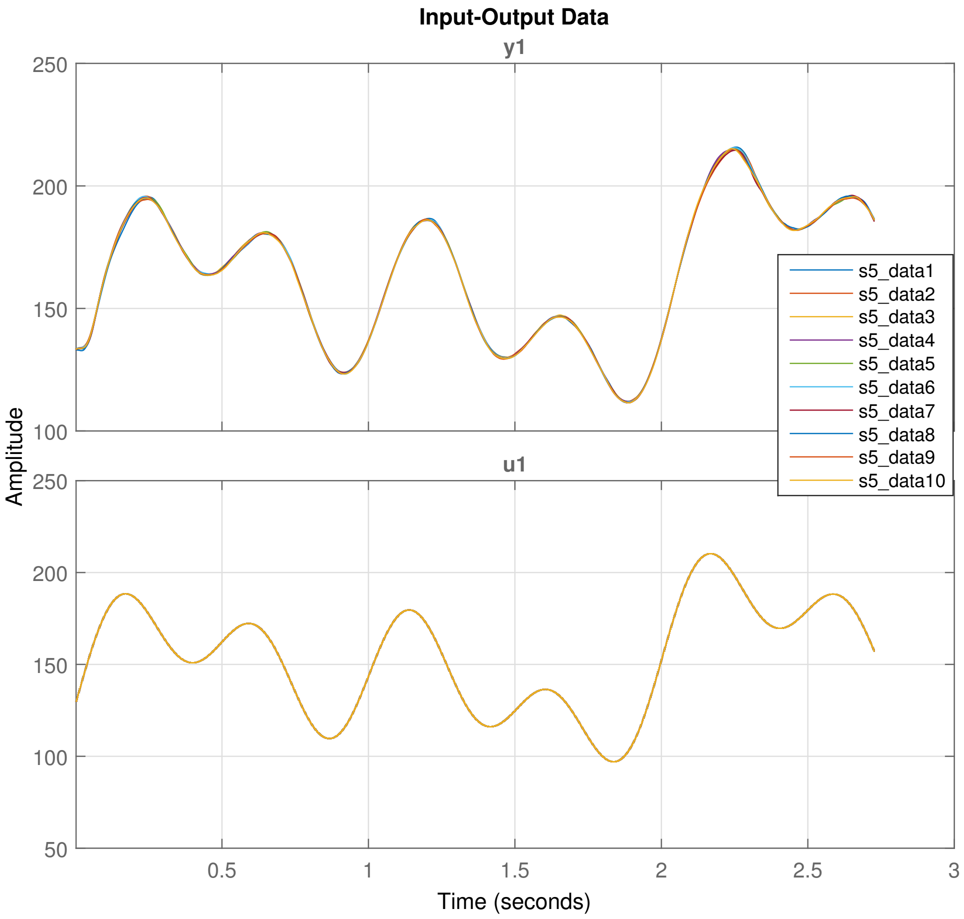


Figure B.5: 10 different strokes with input signal 5.

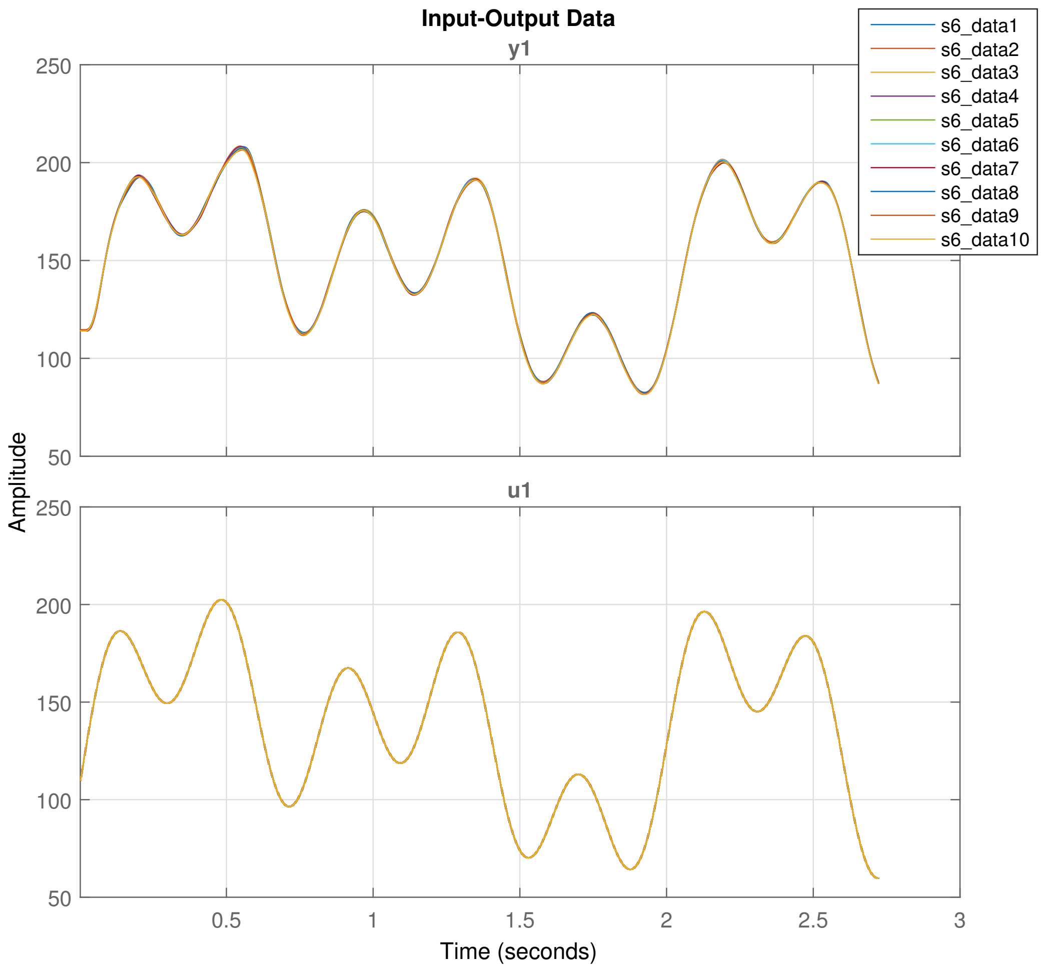


Figure B.6: 10 different strokes with input signal 6.

C

Input/output data second collection

The following input output data were used to build the plant model in the final test described in Chapter 6.

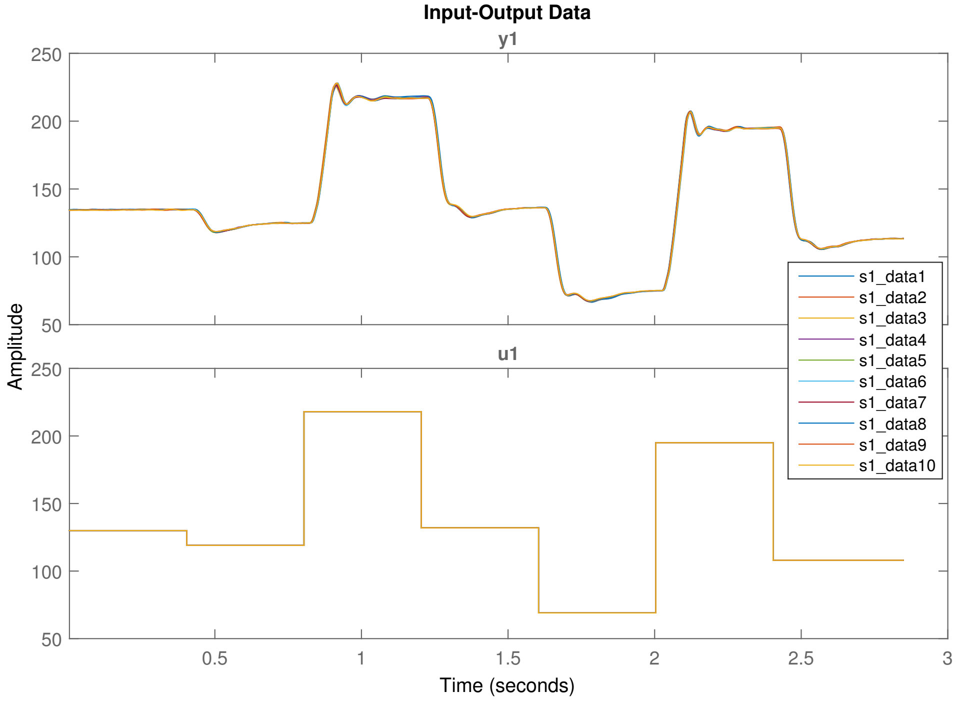


Figure C.1: 10 different strokes with input signal 1.

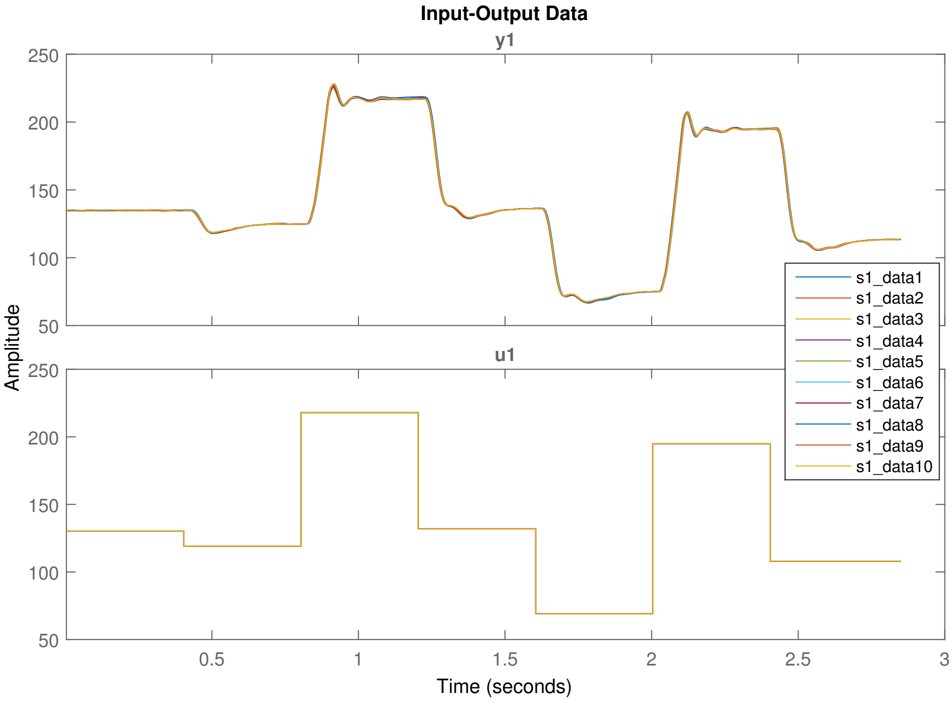


Figure C.2: 10 different strokes with input signal 2.