LUT UNIVERSITY
LUT School of Energy Systems
LUT Mechanical Engineering

Jussi Asikainen

DEVELOPMENT OF A UNIVERSAL REAL-TIME CONTROL PLATFORM FOR A HYDRAYLIC SERVO SYSTEM, UTILIZING THE SYSTEM'S MATHEMATICAL MODEL AND MULTIBODY SIMULATION

Examiners: Professor Heikki Handroos
D. Sc. (Tech.) Hamid Roozbahani

TIIVISTELMÄ

LUT Yliopisto
LUT School of Energy Systems
LUT Kone

Jussi Asikainen

Reaaliaikaisen simulaatioalustan rakentaminen hyrdauliservojärjestelmälle käyttämällä matemaattista mallia, sekä monikappalesimulointia

Diplomityö

2020

60 sivua, 35 kuvaa, 2 taulukkoa ja 6 liitettä

Tarkastajat: Professori Heikki Handroos
TkT Hamid Roozbahani

Hakusanat: Reaaliaikainen ohjaus, Hydrauliservojärjestelmä, Matemaattinen mallinnus, Monikappalesimulaatio, Säätötekniikka

Työssä suoritetaan hydrauliservojärjestelmän simulaatiomallin rakentaminen kahdella eri ohjelmalla. Simulaatiomallit yhdistetään samalle simulaatioalustalle, missä niiden toimintaa testataan, sekä optimoidaan erilaisilla säätötekniikassa käytettävillä menetelmillä. Tässä työssä säätimiksi valikoituivat sumea logiikka, geneettisellä algoritmilla optimoitu PID-säädin, sekä PID- säädin.

Simulaatiomallit rakennettiin MATLAB Simulink, sekä Mevea ohjelmistoympäristöihin. Mevea malli yhdistettiin MATLAB Simulink:iin, jotta molemmat mallit toimisivat samalla alustalla ja Mevea malliin voitaisiin yhdistää rakennetut säätimet. Työssä esitellään Kuinka molemmat rakennetut simulaatiomallit käyttäytyvät mallinnetulla simulaatioalustalla.

ABSTRACT

LUT University
LUT School of Energy Systems
LUT Mechanical Engineering

Jussi Asikainen

Development of a universal real-time control platform for a hydraulic servo system, utilizing the system's mathematical model and multibody simulation

Master's Thesis

2020

60 pages, 35 figures, 2 tables and 6 appendices

Examiners: Professor Heikki Handroos
D. Sc. (Tech.) Hamid Roozbahani

Keywords: Universal Real-time Control, Hydraulic Servo Systems, Mathematical Modelling, Multibody Simulation, Controller tuning

Goal of the study was to create two digital models from an existing hydraulic servo system in order to find best controlling input to control the real machine. These formed digital models were connected to real-time control platform. Testing with real machine was not possible due to faced problems so mathematical models were only compared by each other. Controlling methods that were selected for this study were fuzzy controller, Genetic algorithm tuned PID- controller and PID- controller.

Digital models were build using MATLAB Simulink and Mevea simulation programs. These programs use slightly different approaches for consisting hydraulic servo system. Simulink uses known mathematical formulas and theorems, where Mevea uses these same theorems but more sophisticated.

Mevea based model was connected to Simulink using an interface, which allows to import and export data from Mevea to Simulink. Using this connection between Mevea and Simulink it was possible to control Mevea using Simulink based controlling methods. The results of both systems behavior are presented as results with different point of views how systems differ from each other.

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

a_1	Coefficient of effective bulk modulus
A1	Cylinder area
a_2	Coefficient of effective bulk modulus
A2	Piston side area
a_3	Coefficient of effective bulk modulus
β_i	Effective bulk modulus of cylinder
c_s	Flow constant
DE	Differential Evolution
E_{max}	Young's modulus
F_c	Coulomb friction
F_f	Friction force
F_s	Static friction force
FIS	Fuzzy Interfere System
GA	Genetic algorithm
$g(\dot{x}_p)$	Steady-state characteristics
IAE	Integral of Absolute Error
IE	Integral of Error
ISE	Integral Square Error
ISTE	Integral Square time Error
ITAE	Integral time Absolute Error
ITE	Integral time Error
ITSE	Integral time Squared Error
k	Gain
k_v	Viscous friction
L_i	Laminar leakage flow coefficient
L_1	Laminar leakage flow coefficient
L_2	Laminar leakage flow coefficient
LQ	Linear Quadratic
m	Mass
p_1	Cylinder side pressure

p_2	Piston side pressure
p_{max}	Maximum pressure
p_s	Supply pressure
p_t	Tank pressure
Q_1	Valve flow cylinder side
Q_2	Valve flow piston side
Q_{L1}	External leakage cylinder side
Q_{L2}	External leakage piston side
Q_{Li}	Internal leakage
σ_0	Stiffness coefficient
σ_1	Damping coefficient
u	Input signal
u_s	Valve’s signal
v_1	Cylinder side pipeline volume
V_1	Cylinder volume
v_2	Piston side pipeline volume
V_2	Piston side pipeline volume
v_s	Stribeck velocity
x_p	Piston position
z	Deflection of contacting parts
ζ	Damping ratio

1 INTRODUCTION

The need of building up real machines in first place and testing them on site have been decreasing since digital approaches have showed their potentiality by lowering costs for example in 3D and control designs. Digital approaches include concepts such as digital twins, simulation models and real-time simulation, which includes all the parts that real machine would be consist of. (Belanger, Venne & Paquin. 2010. P. 37.) The use of these digital approaches gives possibility to modify and optimize already existing machines or lines, or for example to predict how machines operates during their life-cycle. For example, digital twin approach is used in predictive maintenance to detect failures in system before they occur. Digital twins can be used to build simulation models as well, which are used to train operators without the need of an actual operating machine. (Qi, Tao, Ying & Zhao. 2019. p. 237–238.)

The approach to conduct digital twin depends on application and used software, where some of the softwares requires building up functions within them, and some has built in libraries for functions and components. One of these softwares is MEVEA, which is used in this thesis as well. MEVEA is simulation software that provides real-time simulation platform for building up simulation model. It has multiple libraries for building up hydraulics or power transmission part and many other within it. Mevea can be combined to other softwares like Python or MATLAB Simulink for plotting results or building up controllers. (Mevea. 2020a.)

1.1 Background and scope

Goal of this study is to build controlling platform for real-time simulation, where in theory any kind of controlling system can be implemented to. The controlling platform is tested using two build up digital models from same real system. Digital models are build using Mevea simulation platform and MATLAB Simulink. Different controlling method and different optimization methods are implemented to both machines. Idea of building these models is to find best controller and controlling input by running the models by first using MATLAB Simulink and Mevea and then to feed the control input to real system. One approach is as well to compare how accurate these built up models are, because they use slightly different approaches to configure the system. Simulink uses only known formulas and theorems as numerical values, where MEVEA has built in parameters for multiple components. The system itself is located at LUT Lab of Intelligent Machines and more detailed information about the system is presented in chapter "Hydraulic Servo System". Various researchers have already published research about the same machine and for example Simulink model from this same system have been built by many researchers. How this research differs to previously publications is presented in chapter 1.4 "previous studies".

Different controlling and optimization methods that have been chosen for this study are: PID- controller, Genetic Algorithm (GA) tuned PID- controller and fuzzy logic controller. These controlling and optimization methods uses slightly different approaches. After both of the digital models are built, they are implemented to same controlling platform, where digital models form the controlling input and controller parameters for the real machine. Concept of combining these three models are shown in figure 1.

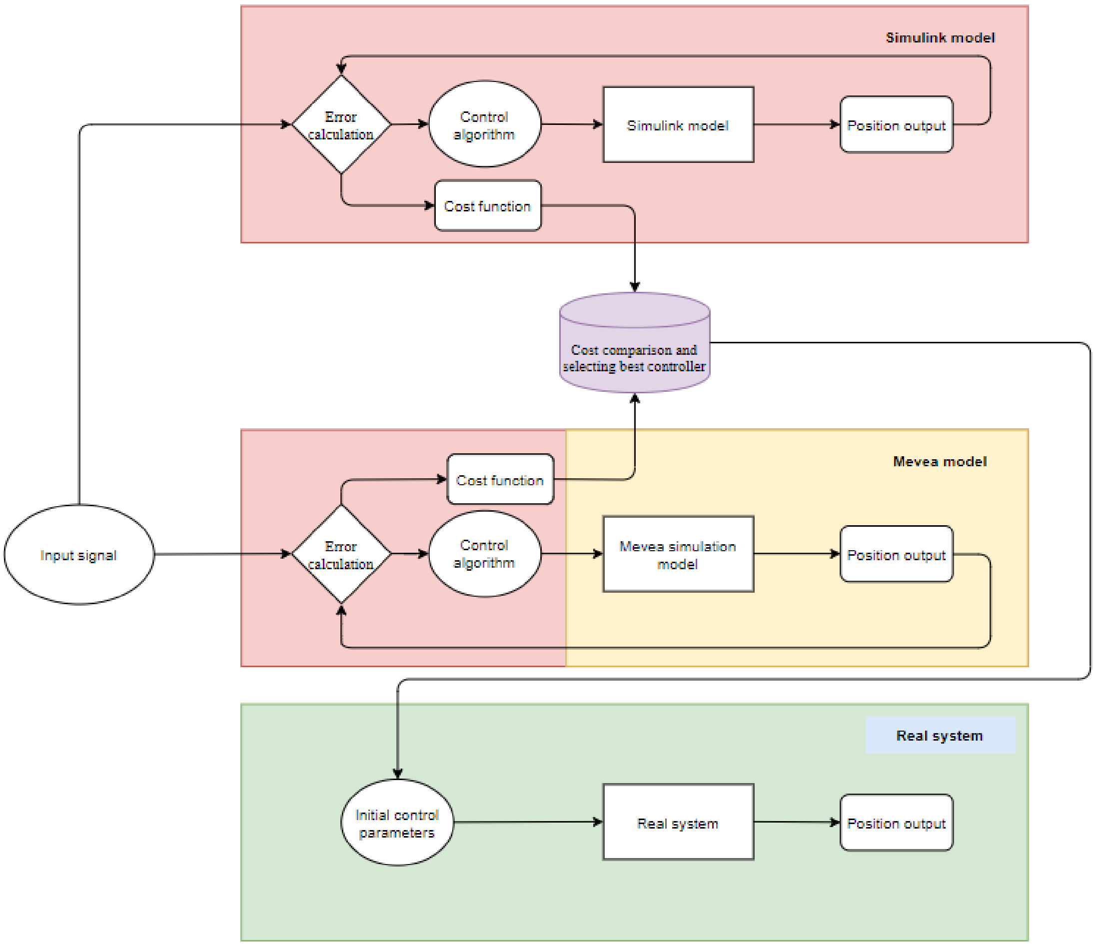


Figure 1. Topology of the consisted controlling platform.

Formed digital models are used for comparing different controlling and optimization methods for finding the best controller and system input for the real system. Both Mevea and Simulink models are executed simultaneously and built up cost comparison of error between output and input determines, should either Mevea or Simulink model input be fed to real system. Where Mevea simulation software is used to only built up mechanisms and hydraulics, input signal to Mevea model must be fed from MATLAB Simulink Connection between model is formed using Simulink external interface that Mevea oy provides. Output signal from the position movement is fed back to MATLAB Simulink from Mevea for cost comparison and control error. This cost comparison compares how much error Simulink and Mevea models have configured and then selects better signal that should be fed to real system. In figure 1 parts that are modelled in MATLAB Simulink environment are described using red boxes. Mevea model is described using yellow box and real system by green box.

1.2 Research questions

From previously described aspects the research questions for this thesis can be formed as following:

- How to construct working Mevea and Simulink models?
- How to generate input signal in Simulink and use it to control similar Mevea model?
- How to create different controllers and test them in both systems?

Based on these research questions, the timetable, or the order how problems must be tackled to able to complete the research can be formed. The task list can be formed as following:

1. Conduct literature review from the topic.
2. Develop Simulink based model of the hydraulic servo system.
3. Develop Mevea based model of the hydraulic servo system.
4. Test both models with manually tuned PID- controller.
5. Develop different controlling and optimization methods.
6. Test the real system using different controllers and input signals that are created in Simulink or Mevea.

By following these 6 steps and answering to the research questions, research should be completed and cover wide area of different aspects in simulation models, hydraulic systems and controlling methods.

1.3 Structure of the thesis

Structure of this thesis follows the basic research structure that divides research to five different parts, which are: Introduction, methods, results, and discussions (IMRAD). This research follows IMRAD structure as well, where chapters are divided to Introduction, Literature review, mathematical models, results, discussion, and conclusions.

In introduction chapter the motivation of this thesis is presented, together with research goals, research questions and task list that works as a roadmap for completing the thesis. Introduction chapter creates the base for introducing the problem that is faced, introducing for example the real system where from mathematical models are formed. In literature review necessary information is conducted for researcher and for readers to be able to understand what is researched and how it is done. In mathematical model's chapter equations that works as base for Simulink model are presented. In same chapter the modelling phases of Mevea model are also described step by step. In mathematical model's chapter is presented as well other necessary things, like how controllers are formed and how models are combined with each other. In first three chapter the base is formed for final three chapters that considers with presentation of the results, discussion about the research and conclusion. In conclusion is evaluated how well did the research went by comparing final results to initial plan, that was formed before this research was started.

1.4 Previous studies

Controlling platforms, servo hydraulic systems, simulation models and digital twins are widely surveyed topics, and there are two research that are related directly to this thesis. Both of those researches contain necessary information that is widely used in this research as well, and without the help of previous researches this research would have been impossible to keep in the frames of master's thesis. Researchers Doctor of Science Hamid Roozbahani, Professor Heikki Handroos and Huapeng Wu have surveyed the same system in their research "Robust adaptive control of a hydraulic servo system by utilizing real-time simulation". Doctor of Science Hassan Yousefi used the same system that is surveyed in this research to conduct his research when pursuing for title of Doctor of Science. Hassan Yousefi research's title was "On Modelling, System Identification and Control of Servo-Systems with a Flexible Load."

In the research "Robust adaptive control of a hydraulic servo system by utilizing real-time simulation" same model with more accurate leakage model is built as it is built in this thesis using Simulink. In that research PID- controller was optimized using Particle Swarm Optimization (PSO), together with the behavior of E coli bacteria. Built up simulation model was connected to control platform that enables to run real-time simulation and operate the real system at the same time. The control platform was used to compare systems inputs and outputs using different controllers. Yousefi Hassan used the same Simulink model in his doctoral research, but he used fuzzy gain-scheduling controller for controlling the model. Hassan did not perform real-time simulation, but he used more sophisticated methods to optimize behavior of the system. He used methods suchs as Differential Algorithms (DE) and neural network.

Manouchehr Mohammadi build a real time simulation from an excavator using MeVEA simulation platform. In the research Mohammadi used connection between Excel and MeVEA with the built-up python code that forms real time connected bridge between Excel and Mevea. Connection between Excel and MeVEA can be seen similar as the connection is built in this research between MATLAB Simulink and MeVea. (Mohammadi. 2017.)

This research uses the same formulas that are presented in previously mentioned researches and therefore information that this research contains is presented in those researches as well. This research is the first time where a Mevea model is formed from the surveyed system. Other researchers used Simulink to control and optimize the real system. Those researches used more sophisticated methods for optimizing controlling inputs, but in the end the bottom frame of all these researches remains almost the same. More about simulation and its usage is presented in chapter 2.7 "simulation process".

1.5 Hydraulic servo system

The hydraulic servo system from where mathematical models using MATLAB Simulink and Mevea are constructed is located at LUT Lab of intelligent machines. This system is presented in figure 2 below.

Figure 2. Hydraulic servo system.

This system consists of servo valve, hydraulic cylinder, mass, data acquisition system, hydraulic circuit, sensors and electrical convertors and amplifiers. Systems servo valve is Bosch Rexroth 4WRPET 6 directional control valve (DVC), with nominal flow rate of 40 liters per minute. Four in 4/3 DVC means that the valve has four different ports: A = cylinder side, B = piston side, P = Pump and T = tank. Number three comes from that valve has three different possible positions:

1. Flow paths from P to A and B to T.
2. Flow paths from P to B and A to T
3. No flow paths, all ports are closed.

All the 4/3 DVC valve parameters are shown in catalog that is shown in appendix I. Input voltage that is fed to valve's Linear Variable Differential Transducer (LVDT) provides control signal, that is used to control valve's spool displacement. This spool displacement determines, which flow path occurs. Spool displacement is used to determine valve flow rates, which are used to calculate piston and cylinder side pressures. These cylinder and piston side pressures are measured using pressure sensor. Pump pressure, P_t = 14 MPa and tank pressure P_t = 0.3 MPa are assumed to be constants during simulation. Valve's output signal is restricted to ± 10 voltages. Valve flows provides the movement to hydraulic cylinder, which has maximum stroke of 1 meter and bulk modulus of 1.8 GPa. Hydraulic cylinder is connected to load, which movement is measured by another LVDT position sensor. Data acquisition system in the real system is executed using dSPACE – DS1103. This dSPACE signal processor process input signal that is fed to valve. All the sensors and actuators in this hydraulic servo systems are connected to each other using computer.

1.6 Controlling methods

Where PID controller is the most common form of controllers in modern industry, are fuzzy logic and especially GA more rarely used approaches. In figure 3 is presented the principles of fuzzy logics.

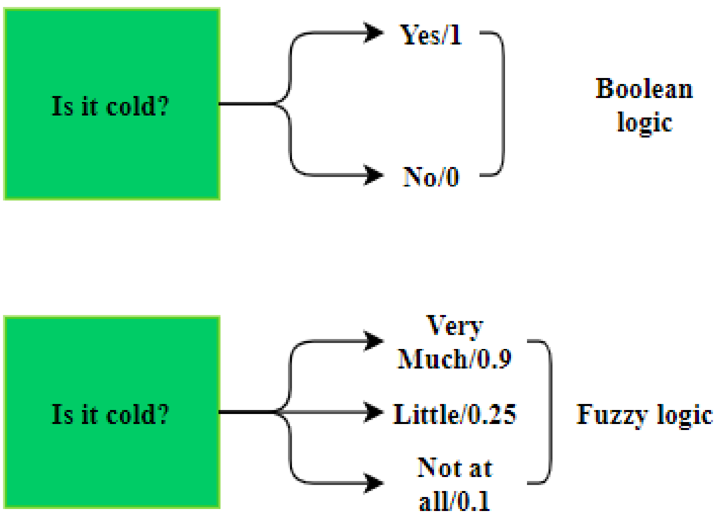


Figure 3. Fuzzy logic principle. (Brain4ce Education Solutions, 2020).

Fuzzy logic evaluates the problem by providing values between maximum and minimum values of the problem, depending how close the solution is. In figure 1 cold is a measured value from 0 to 1 (from warm to cold), but values for evaluation in fuzzy logic can be any numerical values. Comparing fuzzy logic to Boolean logic, which provides only values 0 and 1 to define how cold it is, does fuzzy logic provide much more accurate value for the temperature. Fuzzy logic was first presented by Professor Lofti Zadeh in 1965, who wanted to present systems behavior more accurately, comparing to Boolean Logic. (Zadeh. 1965. P. 339–340.)

Genetic algorithm was first introduced by John Holland in 1975 in his book Adaptiation in Natural and Artificial Systems. The motivation behind invention of GA's was nature capability to adapt and evolve to fit into new environment. Basically, in nature offspring receive different characteristics compared to their parents. Received characteristics are recombined using mutation and crossover. How well new offspring fits to its environment is determined by evaluating its fitness over objective function. In nature evaluation of fitness over objective function could be seen as natural selection. How this GA cycle and fuzzy logic can be used for designing and optimizing control systems is presented in literature review and research methods in this research. (Reeves. 2010. P. 56–57.)

2 LITERATURE REVIEW

Goal of the literature review in this research is to open up how modelled hydraulic servo system works, and which theorems are related to it. Theorems like closed-loop system, controller optimization and factors that should be considered while optimizing controllers. Controllers are modelled in MATLAB Simulink and steps of modelling different controllers is presented in chapter after literature review.

2.1 Closed-loop system control

All the machines and processes suffer from disturbances that affect to their operation. Using a controller unit to control closed loop system the effect of disturbances can be almost neglected. Closed loop systems use sensors that measures the output of the system, to be able to neglect the effect of disturbances from the process. For example, when considering a temperature control of a fridge user determines the desired value for inner temperature and someone opens the fridge constantly. Sensor is located in the fridge to provide information about the inner temperature that has been changed when someone opened the fridge and let warmer air to flow into it. Sensor provides this temperature change and adds it to original controlling input. This input is fed to the controller and it provides a new controlling input to maintain the desired temperature of the fridge. Schematics from the closed-loop system controlling is shown in figure 4 below. (Parr 2006. p. 1–4.)

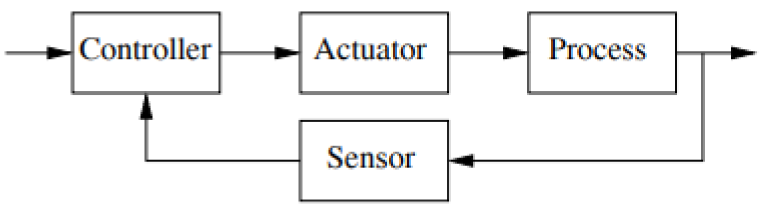


Figure 4. Closed-loop system. (Parr 2006, P. 2).

Controller by itself does not provide desired output from the process, because the controller must be tuned in the way that it gives desired output from the process. There are different factors to consider, when tuning a controller, but the main goal is to find sufficient movement for the exact process. That is why there is no correct way of tuning a controller, only theorems and methods to tune them more effectively. Most common factors to consider, when tuning a controller are presented in figure 5.

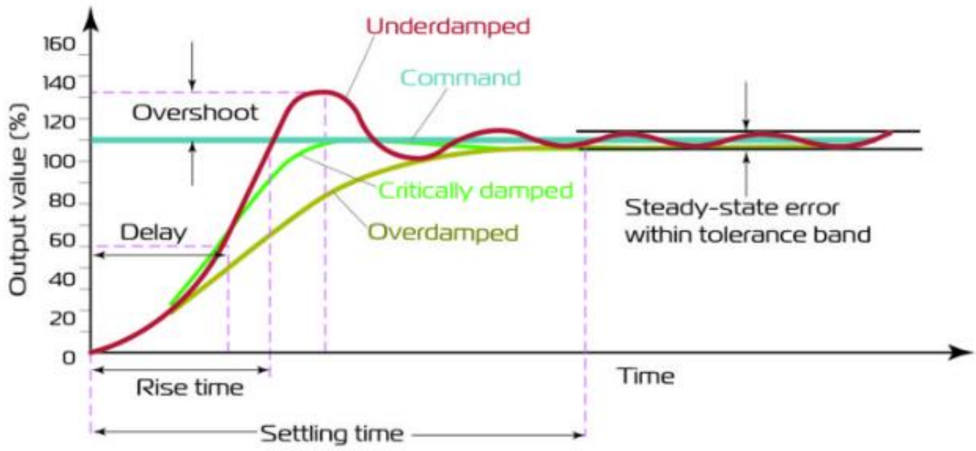


Figure 5. Systems response to step input. (Mathworks 2020).

In the figure blue line presents the input signal of the system, where other lines present the output of the system. Using different optimization methods, for example the overshoot can be neglected. At the same time, the rise time may become larger. (Alfaro & Vilanova 2016, p. 1–2; Visioli 2006, p. 2–3.)

2.2 PID- controller

PID- controllers are the most used controller in automatic processes in nowadays industry applications. When compared to other controllers that are built in this thesis, implementation of PID- controller to actual process is the easiest way to provide reliable automated control of a machine. There are many forms of PID- controller transfer functions in closed-loop system, but the most common form and the one that is used in this research is following:

$$(u_s = k_p + \frac{K_i}{s} + K_d * s) \quad (1)$$

Where k_p is proportional gain, k_i is integral gain and k_d is derivative gain. The parameters of PID- controller depends on the application, for example more fast phased system should have smaller rise time and in more accurate systems idea is to get rid of overshoot and steady-state error. How tuning the parameters of PID-controller affect to systems output is presented in table 1, which works as a road map. The table works when tuning a PID- controller by testing how system output corresponds to change of parameters of PID- controller.

Table 1. PID- controller tuning. (Zhong 2006, p. 8).

Response	**Rise Time**	**Overshoot**	**Settling Time**	**S-S Error**
Kp	Decrease	Increase	Minor change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Minor change	Decrease	Decrease	Minor change

This roadmap of tuning PID- controllers may work when working in simulated environment. Tuning the controller in the same way on a real machine, there may occur vibrations and other undesired movement of the machines. Root fault behind that is the simulation models are always simplified versions from the real system and therefore there occur problems. That is why PID- controllers are usually tuned on site (Zhong 2006, p. 1–13.)

2.3 Genetic algorithms

The idea behind genetic algorithm (GA) follows the rules of nature, where the individuals that adapts to surrounding conditions will survive. Optimization process using GA follows this same path, where predetermined condition stops the loop of forming new generations, if condition is achieved. Whole GA optimization loop is presented in figure 6 below. (Gurunathan 2003.)

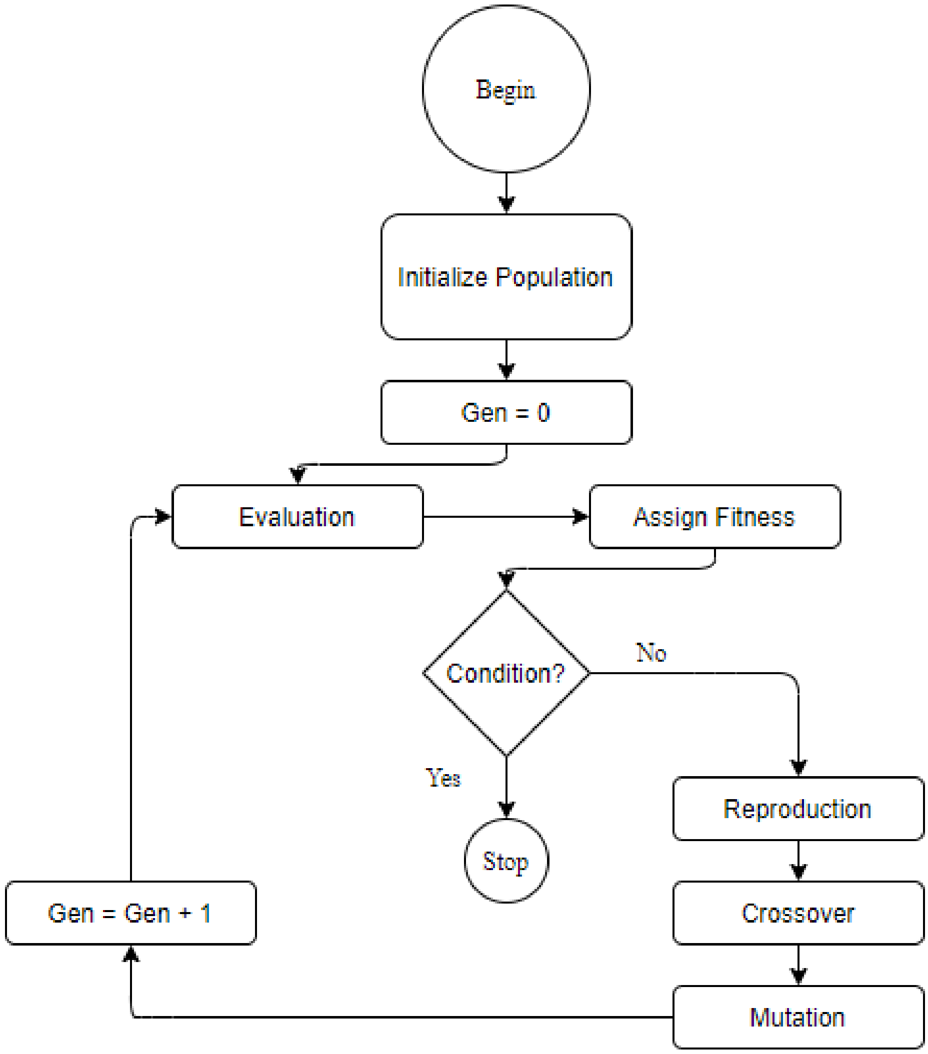


Figure 6. Genetic Algorithm optimization cycle. (Gurunathan 2003).

GA optimization process start by forming initial population, which can be described as generation zero. During one loop of GA optimization formed population is first evaluated using fitness function. This fitness function evaluated how fit formed population is for tackling the problem that GA is formed to optimize. If generation meets the requirements, the GA loop stops here. Otherwise loop continues by reproduction, crossover, and mutation segments, which forms new generation that repeats the GA cycle that was presented. In preproduction stage the most suitable individuals that were previously valued using fitness function are selected to continue for tackling the problem. In crossover a generation is formed from these good individuals that were selected in reproduction. In mutation section formed generation gets its final shape. Sivanandam & Deepa 2008, described the mutation in GA’s as following:” If crossover is supposed to exploit the current solution to find better ones, mutation is supposed to help for the exploration of the whole search space.” After mutation new generation is officially formed and it repeats the cycle. GA optimization cycle keeps running until good enough generation is formed that fits for predetermined conditions. (Sivanandam & Deepa 2008, p. 1–3, 15–16 & 56; Gurunathan 2003.)

2.4 Fuzzy logic

Fuzzy logic was built for presenting the functionality of different problems more specific. In the time before fuzzy logic existed, problems were presented as ones and zeros. For example, A. Q. Ansari presented in his journal following statement: “‘The incandescent bulb glows at a supply voltage of 220V’. According to this statement, the bulb will glow (representing state ‘ONE’) at 220V and not otherwise.” In this statement if the solution was correct, logic gave yes as an answer, which is presented by number one. Otherwise logic provided number zero, which means that the solution was not correct. Implementation of fuzzy logic to mentioned light bulb problem, fuzzy logic would provide values from 0 to 1, depending how close the solution is. The brightness of the bulb would vary depending on the value of fuzzy logic, but it would not be unlit until the voltage reach value of 220. (Ansari, 1998.)

Accuracy of fuzzy set depends on how membership functions determines the problem. Complexity and wanted accuracy are used to determine how many and what kind of membership functions should be used for solving the problem. Sivabandam, Sumathi and Deepa, 2007 presented three following properties that defines fuzzy logics membership functions in their book “Introduction to Fuzzy logic using MATLAB”: Core, Support, and boundary. These three properties of membership functions are presented in figure 7. (Sivanandam, Sumathi & Deepa 2007, P. 73.)

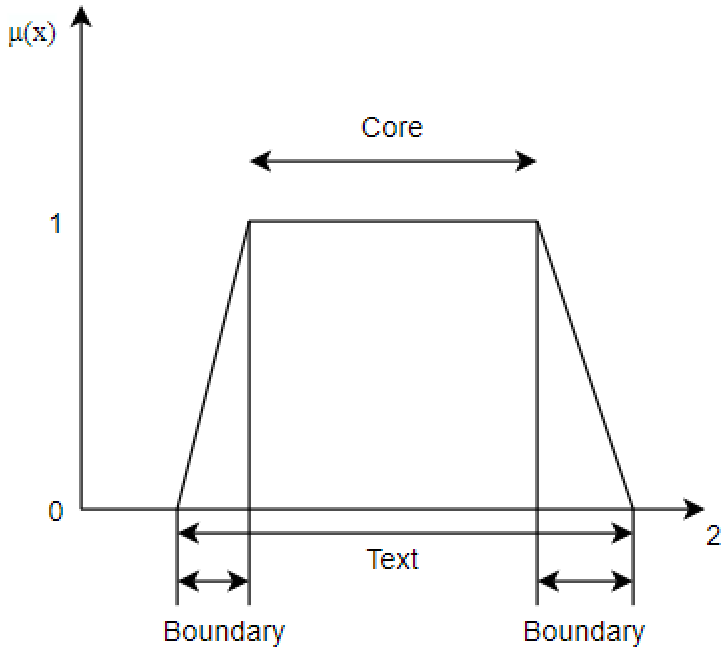


Figure 7. Properties of membership function. (Sivanandam et al. 2007. P. 74)

Core defines the area, where membership function is defined as value 1. Boundaries are used for determining the area, where membership function have value zero, but does not ever reach it highest value. Support determines the area, where membership function get values that are larger than zero. Membership functions are usually combined to multiple membership functions, which demonstrates how close the problem is to its optimum point. Example, from combined membership functions and how they are related to problem is figure 8 below. (Sivanandam et al. 2007, P. 73–74.)

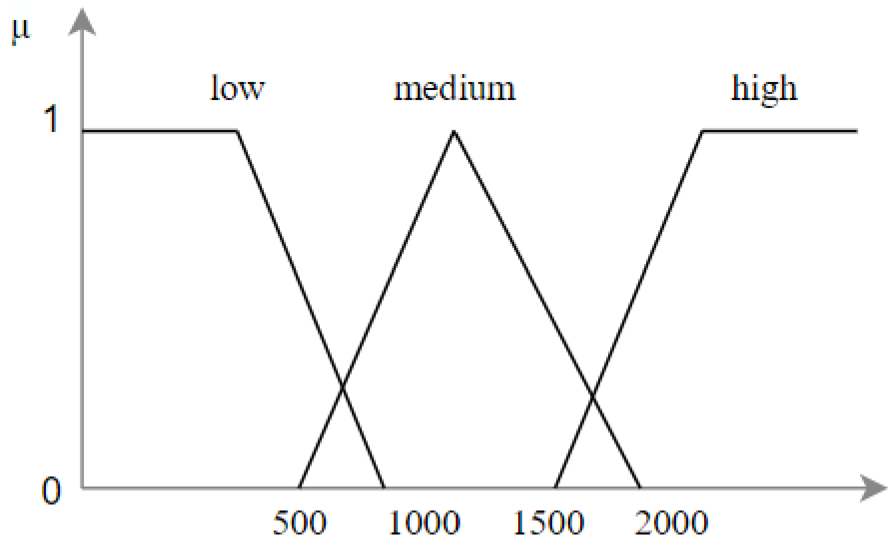


Figure 8. Multiple membership function for defining the problem. (Sivanandam, et al .2007. P. 77).

These membership functions are used to determine numerical values, how correctly they define the problem. Depending on the problem, could figure 9 values present fuzzy logic controlling values for fan speed in revolutions in minute (rpm). This single set of membership function would not work for defining problem, because it would need another set for defining when should the fan rotate faster. For example, when motor temperature reaches 100 degrees should fan rotate at high steep, and when the temperature is below 50 could the fan rotate at low speed. Membership function forms the base in fuzzy logic, for determining rules that finalize how the problem is solved. According to Sivanandam, et al. 2007, the rules of fuzzy logic can be divided to three forms, which are: “

1. Assignment statements,
2. Conditional statements,
3. Unconditional statements.”

Assignment statements are certain values, which are related to problem by using = sign. This relation makes restrictions for solving the problem and may not be functionally for all scenarios. Conditional statements describe the problem by giving certain areas, where the statement get certain value, like in figure 8. Last, unconditional statements describe the problem, without any appropriate conditions. Fuzzy-logic based controller may be combination of these three types of statements. Statements are bound together by set of rules, which uses operators like “or”, “else” and “and” to form rules. Formed membership functions, statements and rules determines how output is form within fuzzy logic controller from the input of the controller. (Sivanandam et al. 2007. P. 113–115)

2.5 Cost function comparison

Cost functions uses different approaches for determining systems behavior. These approaches slightly vary for each other, where other methods determine systems cost by using steady state behavior e(t) can cost function be defined by quadratic cost function as well. (Kumar. P. 2017. P. 8). There are multiple cost comparison methods that displays the systems steady state behavior. Most used of steady state based theorems are integral of square error (ISE), integral of time multiplied square error (ITSE), integral of time multiplied absolute error (ITAE) and integral of absolute error (IAE). These methods define systems performance using following formulas:

$$ISE = \int_0^t e^t(t)dt \tag{2}$$

$$ITSE = \int_0^t te^t(t)dt \tag{3}$$

$$ITAE = \int_0^t t|e(t)|dt \tag{4}$$

$$IAE = \int_0^t |e(t)|dt \tag{5}$$

In these cost functions the limit for time is defined by user. There are other cost functions that uses the steady state behavior for determining the cost of process. Cost functions, such as integral of error (IE), integral of time multiplied by error (ITE), integral of time and square error (ISTE). All of these steady state behavior-based cost comparison methods vary from each other only a bit. (Nuruzzaman M. 2014. P. 110-109). Almabrok, Psarakis & Dounis, 2018 compared the four most used steady state behavior based cost functions shows that square error based cost function has the best response, when using cost function for optimizing systems behavior. In the study ITSE provides the best values, where ISE is second best option, leaving IAE to third and ITAE to worst option. (Almabrok. et al. 2018 P. 7).

Another approach for defining cost function from process behavior is to determine quadratic cost. This method can be implemented only to linear processes, like modelled Simulink model in this research, which is presented using linearization method. Linear Quadratic (LQ) method determines cost of process in following way:

$$J = g(x_t, u_t) = \int_{t_0}^{t} (x_t^2 * Q + u_t^T * R + {x_t}^T * Qx_t)dt \quad (6)$$

In this formula x_t presents output of the system, in this case piston position in x- axis. u_t is the reference input that is fed to system. Q is the infinite horizon of LQ, and it is described as diagonal matrix or in this research as a number one. R and Q are both described as design variables and the ratio of Q divided by R is usually presented, when evaluating the cost of problem by quadratic method. (Murray 2006.)

2.6 Hydraulic servo-systems

Hydraulic servo-system can be seen as normal hydraulic system which main operating tool is servo-controlled valve. Valve can be controlled using DC or AC servo motors and there are different controlling approaches how valve can be controlled. In figure 9 below is presented typical structure of hydraulic servo system.

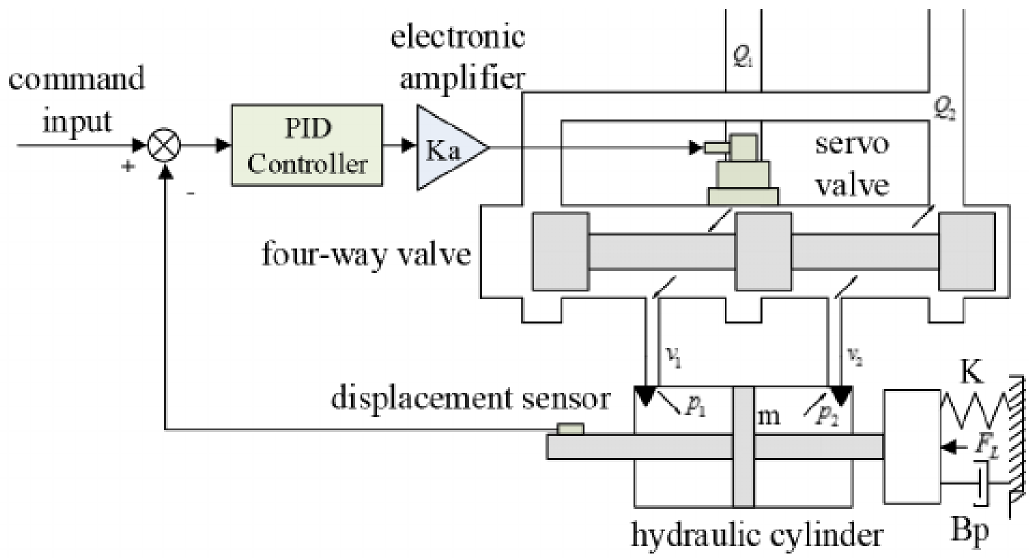


Figure 9. Structure of hydraulic servo system. (Cheng, Lu & Ma 2015).

In the figure x is shown that the main operating tool that control the system is servo valve, which is controlled by PID- controller. Servo valve controls the valve, which operates hydraulic cylinder. Hydraulic cylinder has displacement sensor, which displacement is fed back to PID- controller together with command input. (Cheng et al. 2015). When considering hydraulic servo-system as process in closed-loop control the controlling of hydraulic servo-system may be complex, due to their characteristic to be non-linear. This causes variation within the hydraulic plant operations and therefore hydraulic systems are often linearized before a controlling system is implemented to it. (Plummer & Vaughan 1996.)

When considering the movement of servo valve, servo motors are widely used for power supply in hydraulic circuits due to their advantages compared to fixed-speed motor drives. Servo controlled hydraulic system provides only pressure when needed with high accuracy and fast response time. When servo-controlled valve is combined to hydraulic system capability to produce high forces, the end users have reached up to 90 percentage energy saving compared to fixed speed motors. (Lewotsky 2015.)

2.7 Simulation process

Simulation both offline and real-time simulation has multiple benefits compared to research and development (R&D) that does not use simulation. Could even be said that evaluation of real-time behavior has become essential for multiple complex application, such as design of embedded systems, control of dynamic processes and other data driven engineering applications. Nowadays where simulation programs have become more affordable, more user friendly and the computing power of computers allows to perform complex real-time simulation, have use of both real-time and offline simulation risen. (Popovici & Mosterman 2017. P. 1–2). The bottom line of different simulation approaches remains the same, to model the system in virtual world and use it for optimizing, controlling the process, or training users. Still different scientists have presented different approaches and matters that should be considered when using simulation in R&D. (Belanger et al. 2010. P. 37)

Comer E. R. et al. acquired their patent in 2005 for an idea called "methods and systems for providing simulation-based technical training". One of the ideas that Comer and his team had was to create a data-driven simulation kernel as figure 10 below shows. (Comer. et al. 2005.)

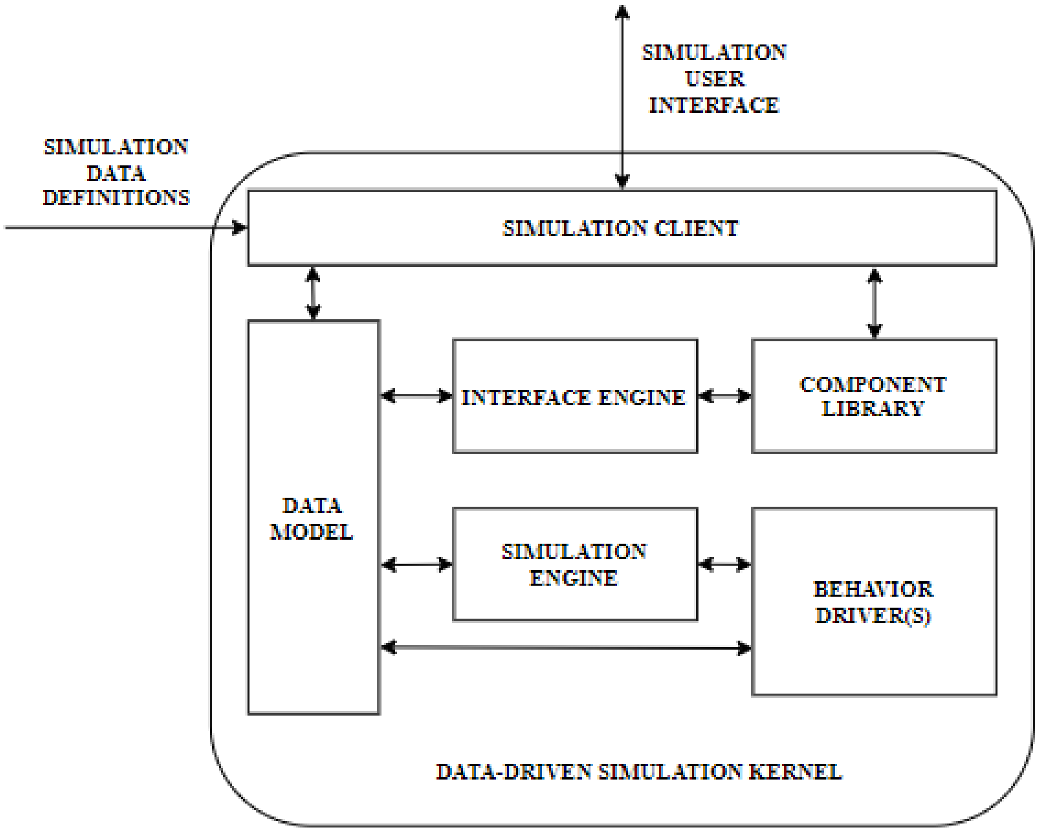


Figure 10. Data-driven simulation program. (Comer et al. 2005).

The idea that Comer and his colleagues had was to acquire patent for a simulation program where the user is only interacting with the simulation interface via the simulation client, which is used for defining parameters for built simulation model within the limits of the simulation program. From the figure above can be see that the team has idea to create built in component libraries for different simulation parts. Like MeVEA has built in hydraulic components and Simulink has different functions that can be dragged and dropped to simulation platform. (Comer et al. 2005.)

Belanger et al. tackles the need of real-time simulation using different simulation methods and model-based design (MBD) approach in their study in 2010. According to that study MBD can be based on “V” chart that is shown in figure 11.

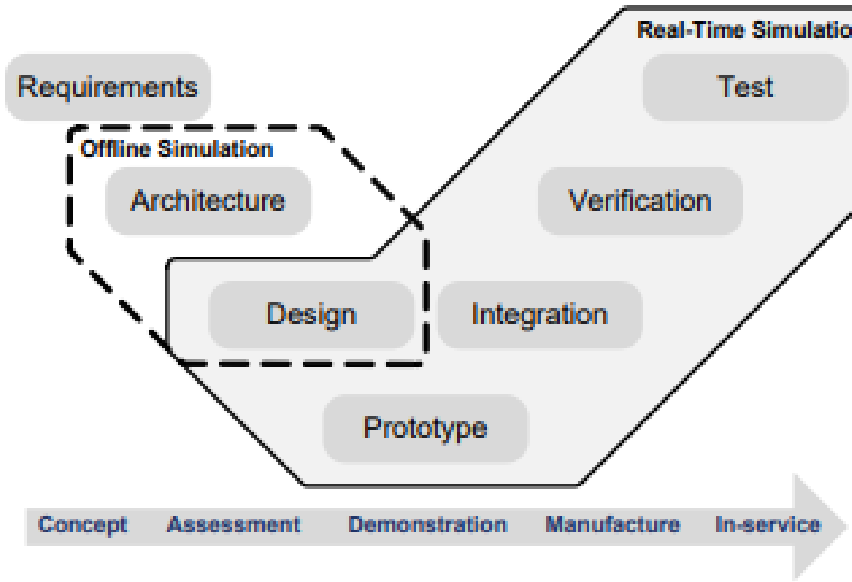


Figure 11. Model-based design. (Belanger et al. 2010, P. 41).

The research mainly considers real-time simulation, but the minimum requirements for offline simulation are shown in figure 11 as well. Where the “V” chart works as a roadmap for engineers to perform real-time simulations in organized manner, does Belanger. et al. presented four necessary steps that combines the use of “V” chart and MBD, which are: “

1. Build the plant model
2. Analyze the plant model and synthetize a controller for it
3. Simulate the plant and controller together and
4. Deploy the controller”

MBD works as approach for building up simulation models, where IMRaD works as roadmap for building up well-structured thesis. Belanger et al. presented one valuable point that should be considered while beginning to build up simulation models, which is different types of simulator interaction. These different interaction types are shown in figure 12 below.

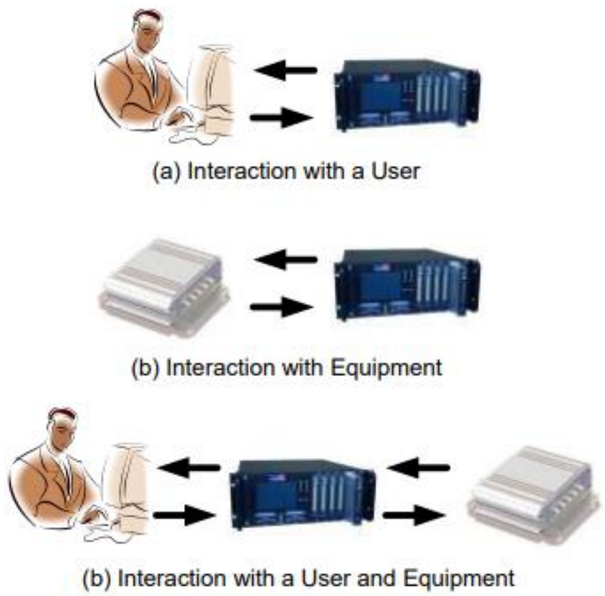


Figure 12. Simulation interaction types. (Belanger et al. 2010. P. 42).

Where the first and last interaction types are working when using MBD approach, where the user has the possibility to modify and follow variable within the process in real-time. These similar interaction types just differ from each other in the way that last interaction type has the possibility to follow the real equipment, where the first interaction type only interacts with hardware. B interaction type is used for following behavior of the equipment, like in predictive maintenance. (Belanger et al. 2010, P. 37–42.)

3 RESEARCH METHODS

Modieddine Jelaili and Andreas Kroll presents in their book the procedure of modelling hydraulic servo-systems in following way: “A mathematical model is constructed from basic physical laws (such as Newton’s laws and equilibrium equations) using and combining available contributions on theoretical modelling of hydraulic systems found in relevant literature. The results if usually a non-linear dynamic (simulation) model of the hydraulic servo-system, including actuator, servo-valve, pipelines, and physical parameters for the given applications.” (Jelaili & Kroll 2012, P. 4.)

Mathematical functions from the real system have already been built by multiple researchers, which are presented in chapter 1.4 “previous studies”. The base of the Mevea model follows these equations and therefore uses some initial values that are presented on the same research. Actuators that are formed in Mevea uses the exact values that manufacturers have provided for the actuators. Controllers are built by combining different theorems that is provided by multiple researchers and educators. Pulse input with amplitude from 0.5 to 0.8, period of 4 secs, with 50 percentage pulse width works as a reference input for all different controllers. This chapter contains only the presentation of different systems and controllers, more analysis about behavior of different controllers is presented in chapter results. This chapter works as research methods of this research as well.

3.1 MEVEA simulation software

In Mevea eBook “The digital Twin at the center of R&D Mevea and company and as a software are presented in following way: “Mevea is a Deep Tech company enabling physics-based Digital Twins of Machines and Vehicles that can analyse and predict the behaviour of their real-life counterparts. By applying the laws of physics to the Digital Twin domain, Mevea’s software and services produce life-like simulations that catalyse pivotal innovations such as autonomous machines and predictive maintenance.” Mevea as a company provides digital twin solutions for companies or just software package, which companies can use for building up their own digital twins. Mevea modelling begins by opening Mevea modeller software, which is included in software package. In Mevea modeller under the help icon are located different manuals that contains information and tips how to use the software. Under that same help icon is located beginner tutorial, which helps you to start you Mevea journey. In Mevea beginner tutorial is presented that user should have “basic knowledge on multi body dynamics, hydraulics, actuators, motors and transmissions components. Understanding of different solvers and the meaning of simulation time-step are also recommended.” Otherwise Mevea is very user-friendly and powerful software for building up digital twins that contains hydraulic parts. (Mevea 2020b.)

3.2 MATLAB Simulink

Mathworks (2020) presents Simulink in following way: “Simulink is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB.” To be able to model more complex systems in Simulink system must be linearized and described using different formulas and mathematical theorems. In this research mathematical functions that are used were obtained from literature, in this case from previous research where this the built-up Simulink model highly leans to. Simulink can be connected to various data sources to perform the calculations as well, like in this research it is connected to Mevea. In that case Simulink is only used for visualization of data. (Mathworks 2020.)

3.3 MEVEA model

MEVEA software only includes tools for building up mechanisms and kinematics. Therefore, modelled operating parts that presents movement in MEVEA model are modelled using SOLIDWORKS 3D CAD software. Input signal to the DVC valve is fed directly from MATLAB Simulink to MEVEA model using connection tools that are presented later in this research on chapter “Mevea Simulink connection”. Rest of the modelling that is required for building up the surveyed system can be made using MEVEA. In this research the modelling of hydraulic servo system can be separated in five different steps, which are presented in figure 13 below:

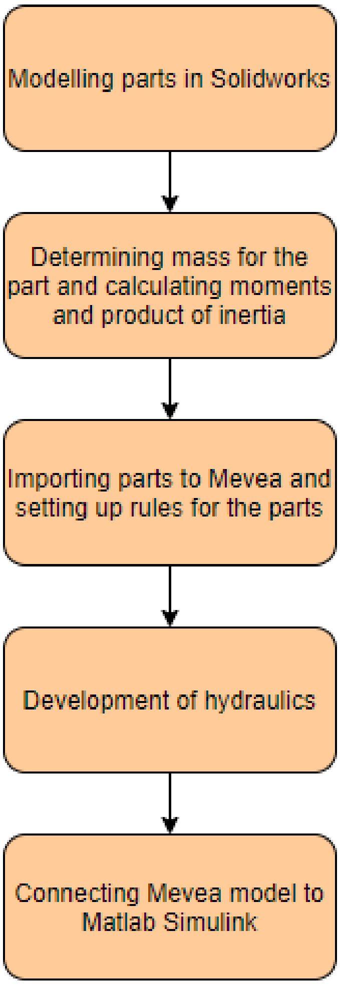


Figure 13. Mevea modelling process.

All the components that are included in MEVEA model are modelled using SOLIDWORKS in the way that simulation model would look like the real system. To model such a system that mimics functionally the real system, following four components must be modelled in SOLIDWORKS: Slide, mass sledge, cylinder, and piston. All the 3D components all imported to MEVEA in STL file format. Slide and mass sledge are imported as bodies and cylinder and piston are presented as dummies in the MEVEA model. STL graphics files can be imported to MEVEA preview window using drag and drop method from selected file source. Dropping the files to MEVEA preview window and selecting correct importing methods MEVEA automatically creates graphics and body or dummy files to MEVEA's model tree from the imported files. These files are used to determine properties of the part, such as position, center of mass, mass, body type, relative body, inertia frame definition and moments and products of inertia. Moments and products of inertia and center of mass are calculated in SOLIDWORKS, by simply checking mass properties of modelled 3D part. All determined properties of four different parts are presented in appendix II. After determining properties of all four parts and redrawing and reloading graphics from the kinematics selection, MEVEA displays the system in the way as it is shown in figure 14 below.

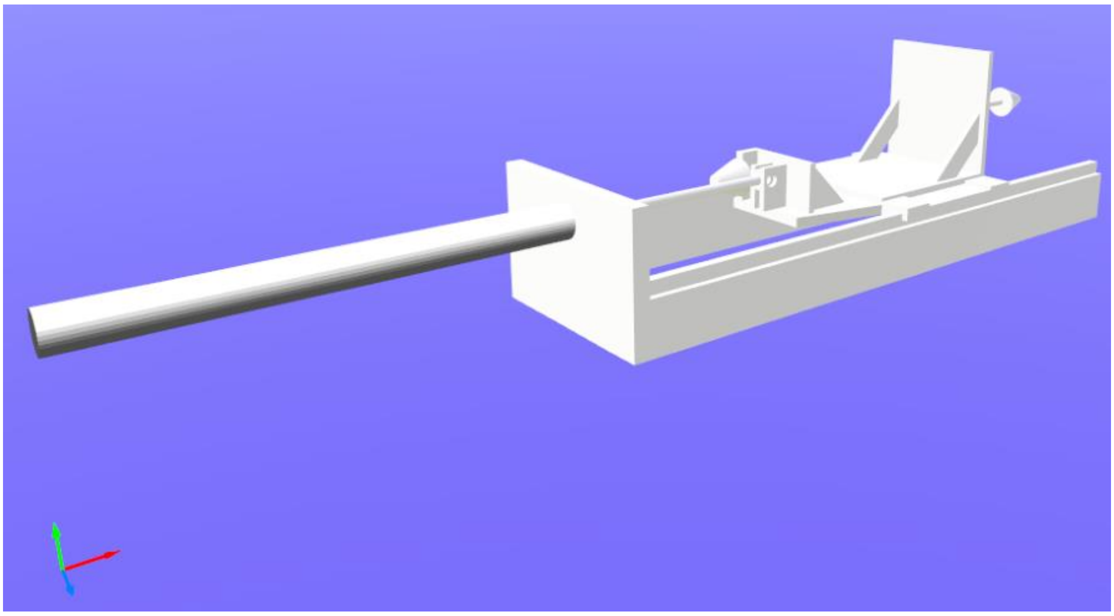


Figure 14. MEVEA model after determining properties of bodies and dummies.

In figure 14, the axis are presented as following: Red presented x-axis, yellow y-axis and blue z-axis. Eventhough model presentation looks correct, it does not have any joints nor inputs that would make the model to mimic the real system. First in modelling process it is better to determine joints correctly, before adding any inputs that produces movement to the model. Orientation of joints in modelled hydraulic servo system is presented in figure 15.

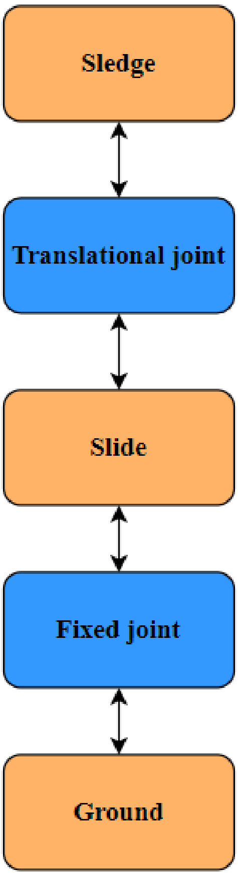


Figure 15. MEVEA model joints topology.

Movement of slide is restricted by fixed joint, which is relative to ground. Translational joint between mass sledge and slide enables mass sledge to move in x-axis direction. Translational joint parameters are set in the way that the movement of mass sledge mimics the movement of real system. Position of joints are done by determining local coordinate systems to bodies. Positions of these local coordinate systems and how constraits are set using these local coordinates are shown in appendix II.

Next force and hydraulic components together with input signal are formed in MEVEA. System has one operating translational force, which location and movement are determined using local coordinates. These local coordinates and parameters of translational force are shown in appendix III. Hydraulic components are formed in MEVEA using initial conditions that are presented in following SIMULINK model chapter. Such as pressures of pump and tank, and cylinder diameters and restrictions are build using given information about the system. Friction is modelled in Mevea using LuGre friction, which uses same parameters that are used in Simulink model for modelling LuGre friction. Parameters of 4/3 DCV are set using values that Bosch have provided from the valve on their valve catalog. Direct link to the valve catalog is located at references of this thesis. After all the hydraulic and force components are build the MEVEA model is ready for building MATLAB SIMULINK connection to import input signal.

3.4 Simulink model

Simulink model from the system is build using known hydraulic equation and derived physic laws by different authors. These equations are formed based on the topology that is presented in figure 16, where actuators and movement of the system is described. In this research leakage flows have been simplified, but rest of the mathematical equations remains the same.

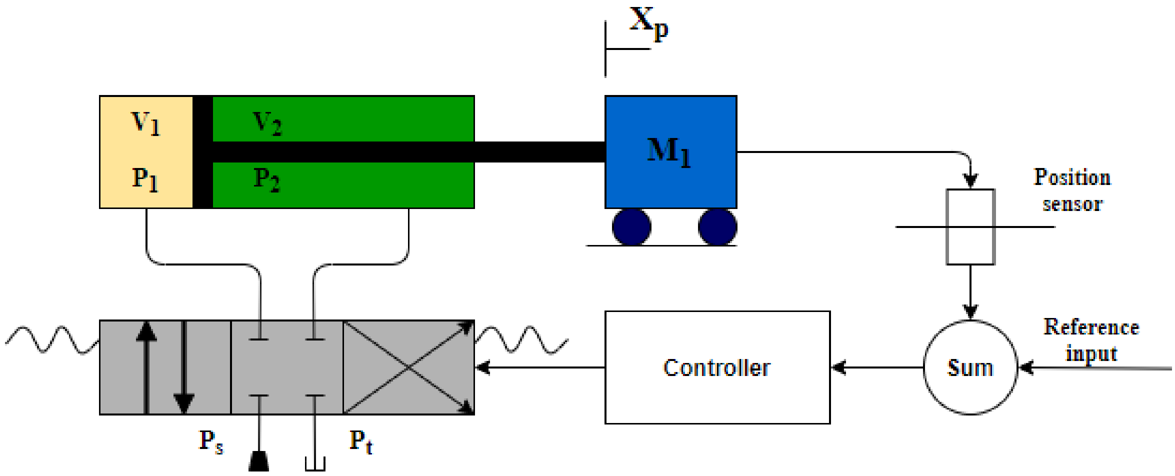


Figure 16. Schematics of modelled servo hydraulic system. (Yousefi. H. 2007. P. 30).

Mathematical equations consist valve dynamics, equation of motion, friction force, valve flows, valve leakages, pressures, and cylinder chamber volumes of the surveyed system. Initial values for all constants that are used in these equations are presented in table 2 below.

Table 2. Initial values for Simulink model.

Constant	Value	Description	Unit
a1	0.3102	Coefficient of effective bulk modules	-
a2	49.18		-
a3	1.843		-
A1	0.804	Cylinder area	m^2
A2	0.424	Piston area	m^2
cs	3.021e-8	Flow constant	$m^3s^{-1}v^{-1}Pa^{-1/2}$
k	0.9907	Gain	-
Li	1.19e-12	Laminar leakage flow coefficient	$m^3s^{-1}Pa^{-1}$
ζ	0.5588	Damping ratio	-
w_n	481.3	Natural angular velocity	Rad/s
Ff	100	Friction force	N
m	210	Mass	kg
v_1	1.07e-4	Pipeline volume	m^2
v_2	1.07e-4	Pipeline volume	m^2
p_s	14	Supply pressure	MPa
p_t	0.3	Tank pressure	MPa
σ_0	1521	Flexibility and damping coefficients of friction force	N/m
σ1	848.3		N/m
k_v	87.74	Viscous friction coefficient	Ns/m
FC	74.81	Coulomb friction level	-
FS	2921	Static friction force level	-
E_{max}	1.8e9	Modulus of elasticity	Pa
P_{max}	28	Maximum pressure	MPa
vs	0.1624	Stribeck velocity	-
L	1	Maximum stroke	m

Using these values and equations that are presented in this following chapter, can the hydraulic servo system be built. Equation of motion (EOM) forms the base forms for the system, which is derived using Newton’s second law. Using Newtown second law EOM becomes following:

$$m.\ddot{x_p} = p_1 \cdot A_1 - p_2 \cdot A_2 - F_f \tag{7}$$

In this equation m is the mass of moving sledge, $\ddot{x}_p$ describes the second derivative of displacement of piston, measured from the zero position. p_1 and p_2 are cylinder and piston side pressures, and A_1 and A_2 are cylinder and piston side areas.

Bosch Rexroth 4WRPET 6 4/3 valve's dynamics are derived in following way:

$$\ddot{u_s} = k \cdot w_n^2 \cdot u - 2 \cdot \zeta \cdot \dot{u_s} - w_n^2 \cdot u_s \tag{8}$$

where k is gain of the system, w_n is natural angular frequency, u is input voltage, ζ describes the damping ratio of the system and u_s signal that is collected from valve's transducer. (Kulakowski B. T, Gardner J. F & Shearer J. L. 2007. P. 94.)

Friction model for the system is computed using LuGre friction model, which considers different frictional variables in the system. Mathematics behind LuGre friction model is presented in three following equations:

$$F_f = \sigma_0 \cdot z + \sigma_1 \cdot \frac{dz}{dt} + k_v \cdot \dot{x_p} \tag{9}$$

$$\frac{dz}{dt} = \dot{x_p} - \frac{|\dot{x}|}{g(\dot{x_p})} \cdot z \tag{10}$$

$$g(\dot{x_p}) = \frac{1}{\sigma_0}\left[F_s + (F_s - F_s) \cdot e^{-\left[\frac{\dot{x_p}}{v_s}\right]}\right] \tag{11}$$

Where F_f describes the friction of moving sledge. σ_0, $\sigma 1$ and k_v are stiffness coefficient, damping coefficient, and viscous friction coefficient. z describes the deflection of contacting parts and $g(\dot{x}_p)$ adapts steady state characteristics. F_s and F_c describes static and coulomb friction and v_s is the Stribeck velocity. (Liu et al. 2014.)

Valve flows for double operating cylinder are determined as flows for both sides of the cylinder. The equations for valve flows are presented in following four equations:

$$Q_1 = c_s \cdot u_s \cdot sign(p_s - p_1)\sqrt{|p_s - p_1|}, when\, u_s \geq 0 \quad (12)$$

$$Q_1 = c_s \cdot u_s \cdot sign(p_1 - p_t)\sqrt{|p_1 - p_t|}, when\, u_s < 0 \quad (13)$$

$$Q_2 = c_s \cdot u_s \cdot sign(p_2 - p_t)\sqrt{|p_2 - p_t|}, when\, u_s \geq 0 \quad (14)$$

$$Q_2 = c_s \cdot u_s \cdot sign(p_s - p_2)\sqrt{|p_s - p_2|}, when\, u_s < 0 \quad (15)$$

In these equations input voltage, $u_{s,}$ which operates the servo solenoid valve determines the direction of flows in valve. p_1 and p_2 are piston and cylinder side pressures and p_t and p_s are supply and tank pressures. c_s describes the flow constant of the valve. (Kulakowski et al. 2007, P. 230-233.)

Valve leakages are divided to internal leakage flow and to laminar leakages, which can be calculated in following way:

$$Q_{Li} = L_i \cdot (p_2 - p_1) \quad (16)$$

$$Q_{L1} = L_1 \cdot (p_1 - p_t) \quad (17)$$

$$Q_{L2} = L_2 \cdot (p_2 - p_t) \quad (18)$$

In these equations' L_i presents the internal leakage coefficient and internal leakage Q_{Li} is calculated using formula 11. L_1 and L_2 are laminar leakage flow coefficients, that are used for calculating laminar leakage flows at cylinder and piston side. (Totten 1999. P. 684.)

For determining valve port pressures, effective bulk modules and volumes for both cylinder sides must be calculated first using following formulas:

$$\beta_{ei} = a_1 \cdot E_{max} \cdot log\left[\left(a_2 \cdot \frac{p_i}{p_{max}}\right) + a_3\right] \quad (19)$$

$$v_1 = A_1 \cdot x_p + v_1 \quad (20)$$

$$v_2 = A_2 \cdot \left(L - x_p\right) + v_2 \quad (21)$$

In this equation bulk modules for cylinder and piston sides can be computed by changing pressure, where rest of the equation remains the same. In these equations E_{max} is Young's Modulus, P_{max} is maximum pressure of the system and a_1 – a_3 are coefficients of effective bulk modules. (Totten 1999. P. 257.)

By deriving bulk modules first, can pressures at valve's port computed in following way (Rabie 2009. P. 201.)

$$\frac{dp_1}{dt} = \frac{\beta_{e1}}{V_1}\left(Q_1 - A_1 \cdot x_p^{\cdot} - Q_{Li} - Q_{L1}\right) \quad (22)$$

$$\frac{dp_2}{dt} = \frac{\beta_{e2}}{V_2}\left(-Q_2 - A_1 \cdot \dot{x_p} + Q_{Li} - Q_{L2}\right) \quad (23)$$

When all the equations are build using MATLAB Simulink a plant should be formed by using the given equations. The figure from the plant is presented in figure 17.

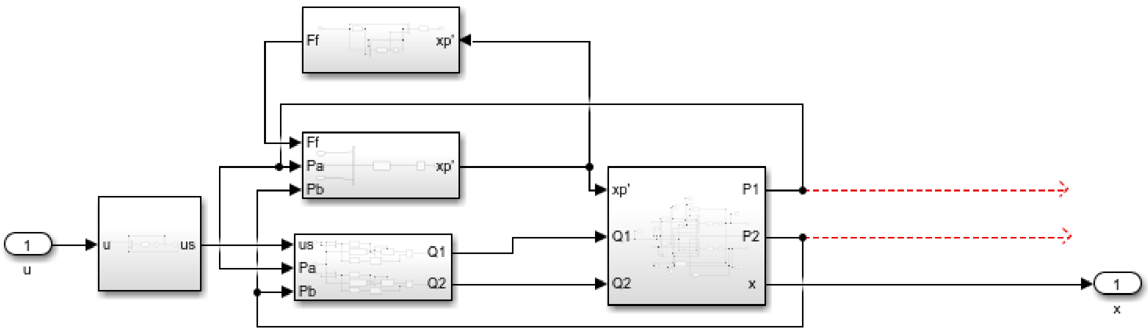


Figure 17. Modelled system in Simulink.

By forming a subsystem from this plant different controlling methods can be implemented to the plant in Simulink. All different controllers are built in same Simulink base, where the system can be tested using different controllers. The whole modelled Simulink with controllers is presented in appendix IV.

3.4.1 Genetic algorithm PID- controller optimization

Genetic algorithm based PID- controller optimization is performed in MATLAB environment. Sanchin Sharma has written a script that is developed to optimize systems PID-controller parameters by just determining systems transfer function. This MATLAB script together with MATLAB's genetic algorithm optimization toolbox optimizes the PID-controller parameters using fitness function, error, and transfer function of plant. The script uses universal form of PID- controller, which is presented in equation 1. When using GA optimization method, first a transfer function from a plant must be formed. In this research transfer function is determined using Simulink linear analysis method. When using this method points must input perturbation and output measurement points be determined to Simulink model. PID- controller is neglected from system in GA optimization, because it is included in optimization MATLAB script. After setting input and output point to Simulink model, configuring of transfer function follows following path: Selecting analysis from Simulink, then clicking control design and selecting linear analysis. This opens the control design window, from where different aspect from the system can be evaluated. In this scenario system is evaluated using results viewer and start bode linearization. Simulink opens results viewer window, where transfer function of the system is presented. Transfer function of system without PID- controller becomes following:

$$G(s) = \frac{6.856e7*s^2+1.184e8*s+2.215e-9}{s^7+545.6*s^6+6.392e5*s^5+5.636e6*s^4+7.998e8*s^3+6.319*s^2+1.368e6*s+1.712e-12} \quad (24)$$

Using this transfer function and MATLAB script that has been written by Sanchin Sharma, PID- controller can be optimized using GA. This MATLAB script is presented in appendix V. From the written script can be seen that it uses step input as an input. Written MATLAB script plots step response of the system in every loop PID- parameter optimization loop, and it works as a good visual tool for determining when PID- controller parameters are good enough to control the system. GA optimization itself is executed using MATLAB's optimization toolbox, which has tools for GA based optimization. This GA optimization toolbox takes fitness function, number of optimized parameters and bounds as an input, rest of the parameters remains default. GA optimization ran for three iterations at it provided following parameters for PID- controller: $K_p = 50$, $K_i = 20$ and $K_d = 1$. Step response plot using these parameters is presented in figure 18 below.

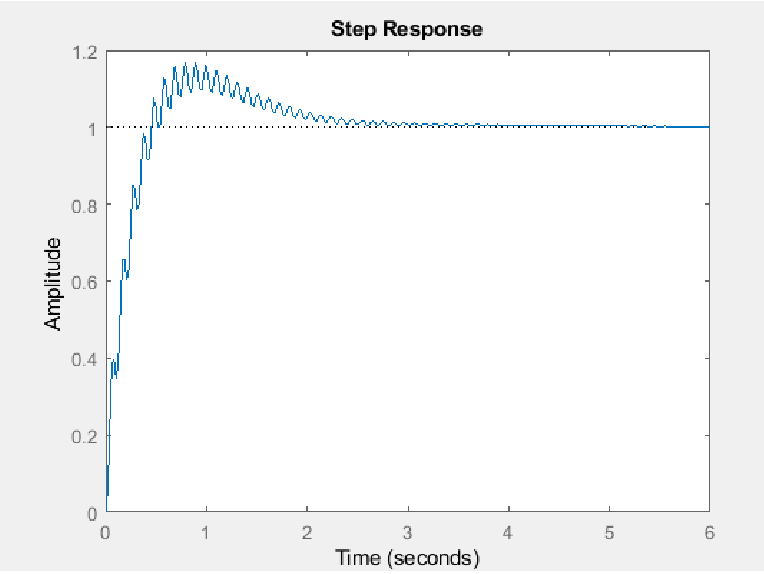


Figure 18. Step response of the system.

Initial value of the response is zero, and its settling time is 3 seconds, with the overshoot of 15 percentages. In Simulink and Mevea control input is formed in the way that system position varies 30 centimeters not 1 meter, so response of the real system using same PID-controller parameters should be better.

3.4.2 Fuzzy logic controller

Fuzzy logic controller parameter optimization is carried out using SISOTOOL command in MATLAB. This command requires system transfer function, which is presented in equation 19. MATLAB SISOTOOL command opens bode, root locus and step response plots using determined transfer function. Determining small enough settling time to step response of the system and finding real zero from root locus, SISOTOOL calculates compensator, which is required for building up boundary conditions for rules in fuzzy controller. Provided value for compensator is 15 and -3 for root locus real zero. Compensator value determines the value of the output from fuzzy logic controller, which is in this case limited to ± 10 voltage.

Fuzzy logic controller in Simulink takes two inputs, control signal and derivative of control signal. Derivative of the control signal works as error signal in this scenario. These two signals are connected to fuzzy logic controller using Simulink mux, which feeds both signals two controller simultaneously (Figure 19).

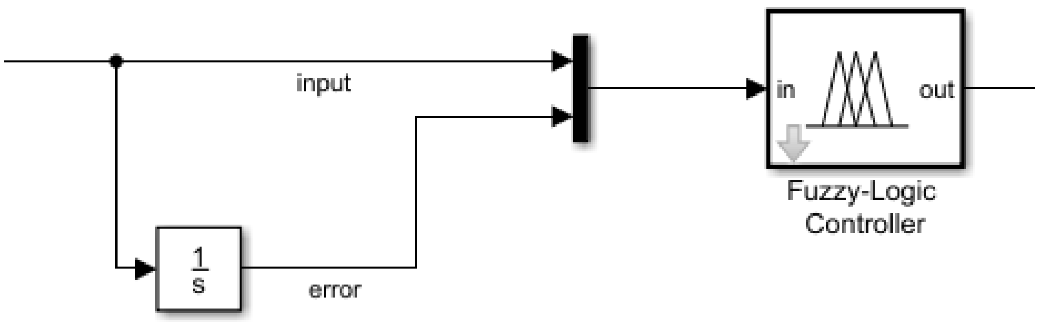


Figure 19. Fuzzy logic controller and its inputs.

Fuzzy logic controller forms the output signal using fuzzy interfere system (FIS) file. This file can be modified by typing fuzzy in MATLAB's command window. In figure 20 is presented fuzzy logic determined rules, membership functions and outputs, which values were obtained by SISOTOOL optimization.

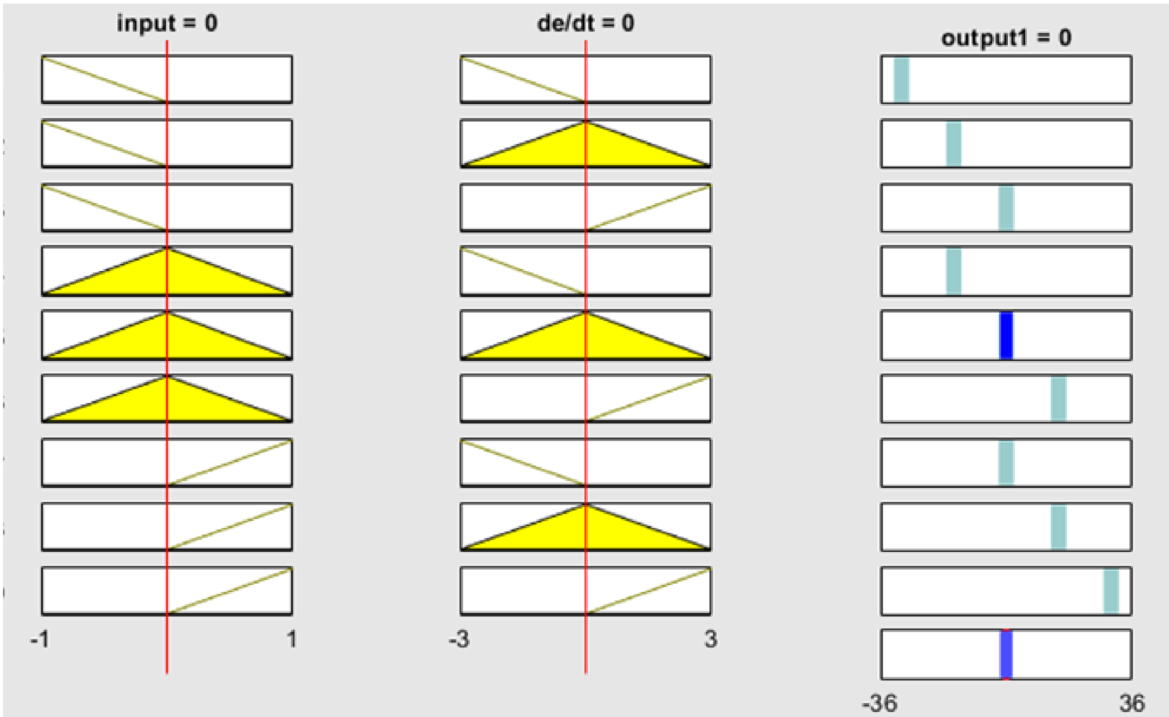


Figure 20. Fuzzy logic membership function and rules.

Fuzzy logic output values depend in which direction the input signal is going and is its value positive or negative. For fuzzy logic controller the input signal is corrected by determining average value for input signal and then subtracting the actual input signal from the average.

The slope of the input signal, which is the error signal gives another rule for fuzzy logic to combine output value for the controller. After determining these values FIS file is ready for importing to fuzzy logic controller. FIS file must be exported to MATLAB's workspace, so the fuzzy logic controller is able to detect it. Then by selecting imported FIS file in Simulink's fuzzy logic controller block, the controller detects the fuzzy file.

3.4.3 Mevea Simulink connection

Transmission Control Protocol/ Internet Protocol (TCP/IP) is used to form connection between Mevea and Simulink. The connection is made using Simulink external interface that is provided directly by Mevea company. The file bundle that contained Simulink external interface for Mevea, included other required files that were coded with C++ programming language. There are three files that are needed to form the connection between Mevea and Simulink, which are: MReceivingdata, MSendingdata and MServerInitializer. These files are used to write the rules for connected signals, such as: time step, ID address that is used for establishing the connection, port number and number of exported and imported signals to Mevea. To be able to create the connection between Mevea and Simulink, all the required files must be in the same working directory, where the Simulink model and MATLAB scripts are located.

Mevea and Simulink are combined using Mevea socket interface that is located at input/ output (I/O) section at the model tree. IP address and port number that creates the connection between software's are determined to this created socket interface. In this research are used IP address 127.0.01 and port number 5111. Six socket signals are created to the socket interface for importing the input signal from controllers to Mevea and to exporting desired values back to Simulink. Variables that are imported back to Simulink are piston and cylinder side pressures and piston position in x-axis. Fed input signal is directly connected from Simulink to 4/3 DVC valve in Mevea. Selected exported signals are exported from Mevea using created data source. Data course creates numerical data from selected variable. Created data source is linked to socket interface that sends the position to Simulink. Simulink external interface where Mevea outputs and inputs are connected in shown appendix VI. The Simulink external interface is created by Jarkko Nokka in 2009. To this Simulink interface same IP address and port number that were determined in Mevea are set to MserverInitializer block. This block configures the time step as well that is used in exported data. Number of connected signals are set to Mreceiving and Msendingdata blocks. Mevea Simulink connection forms the base for creating cost comparison that evaluates, which input signal should be fed to real system. Overview from created Mevea Simulink connection is presented in figure 21.

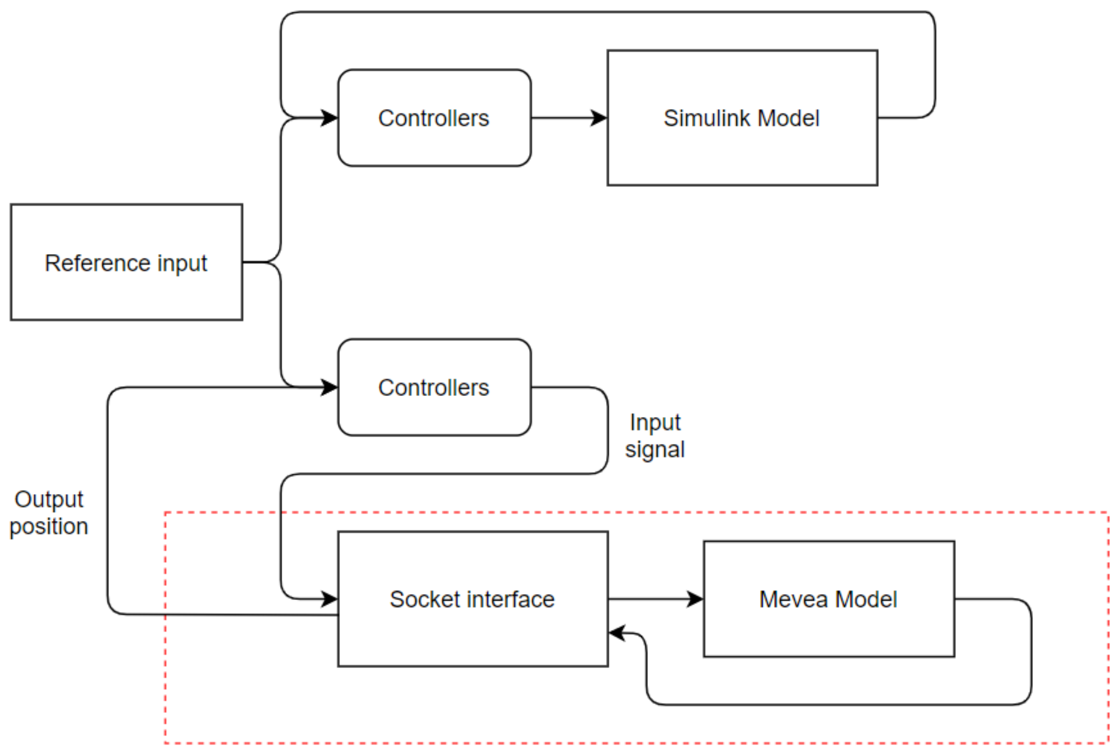


Figure 21. Mevea model connected to Simulink using socket interface.

Mevea operates at the background as its own, but all the data is transferred using Simulink. On figure 19 Mevea forms the parts that are cropped by red box. All the other functions take place in Simulink.

3.4.4 Cost comparison

Cost comparison is built for evaluating the performance of Mevea digital model and Simulink based mathematical model. In this research cost comparison is not used for tuning the controllers, so it does not matter how well different cost comparison methods compare to each other. As shown in literature review, square error based cost comparison methods have provided best results in controller optimization. For this research cost comparison methods that are selected for evaluating behavior of the systems are LQ and IAE. LQ based cost comparison compares the generated error between these models by using cost function that is shown in equation 6. This cost function evaluates the error between reference input and position outputs of both systems. Cost functions is formed in Simulink, by compiling the function that is shown in equation 6 with Simulink blocks. Cost is calculated as well using integral of absolute error (IAE), which formula is shown in equation 5. In this research these two cost comparison methods are used for determining performance of the system using different inputs, which are presented more specific in following "Results and Analysis" chapter.

4 RESULTS AND ANALYSIS

This chapter contains the obtained results that are obtained from testing of formed Mevea and Simulink based simulation of hydraulic servo system. The chapter contains information how formed model differs from the initial concept and how much it affected to obtained results.

In the last version of this research, both Mevea and Simulink models forms closed loop systems with selected controlling methods. When comparing figure 23 to initial concept that is presented in figure 2, can be seen that only the testing with real system is taken out.

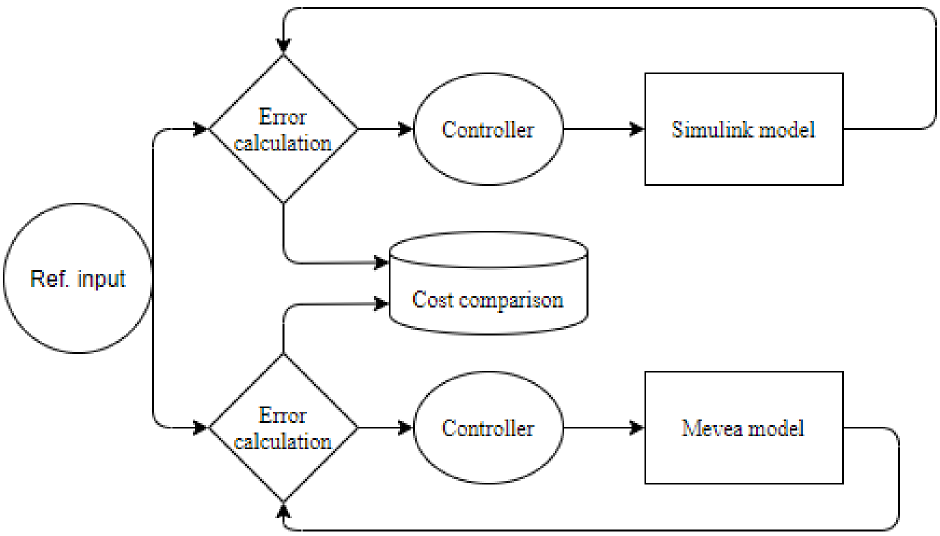


Figure 22. Final concept.

By taking the real system out, it is not possible to get all the results that were presented in introduction. When neglecting real system from this chapter, results considers following aspects:

- Displacement using same controllers in Simulink and Mevea models.
- Displacement using Simulink generated input signal that is fed to Mevea model.
- Comparison between formed models.
- Cost comparison methods

All the results are generated in Simulink. Some variables needed some adjusting, for example Mevea measures cylinder pressures in pascals where Simulink model calculates pressures in megapascals (MPa).

4.1 Starting point

Without controllers the input signal that is fed to both models is the generated reference input. The movement of both models using reference input as fed input signal generates following movement:

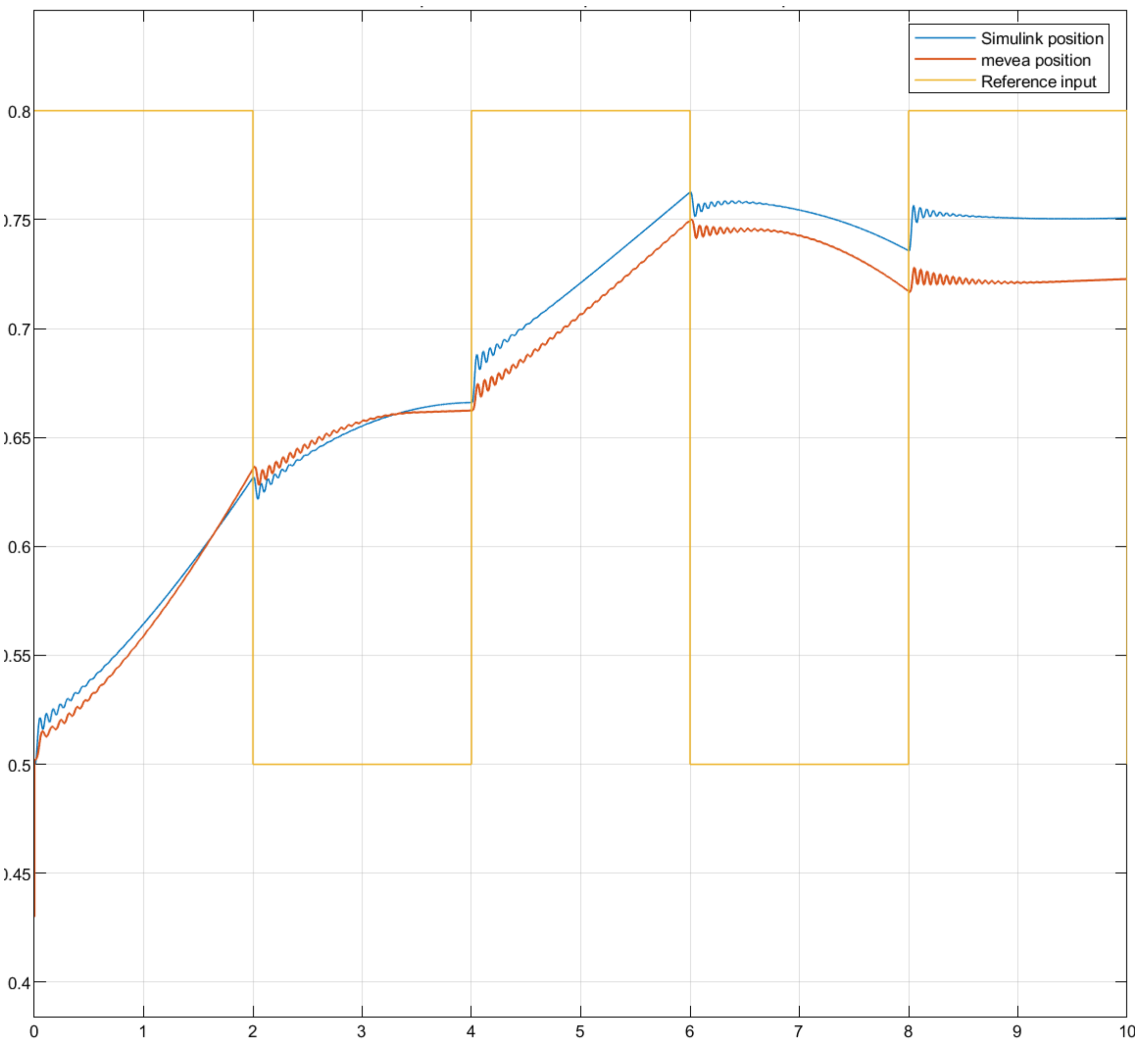


Figure 23. Starting point, before optimizing controller values.

It is clear that properly tuned controllers' effect heavily on behavior of both systems. That is why controllers are optimized using different methods that are presented in chapter 3. These optimizing methods are used to test best controller in Simulink using cost comparison that uses LQ based cost function, that is presented in function 1. Another cost comparison method that is used is IAE, which generated error can be seen in figures 23,25 and 27. LQ based cost comparison is presented in figure 25:

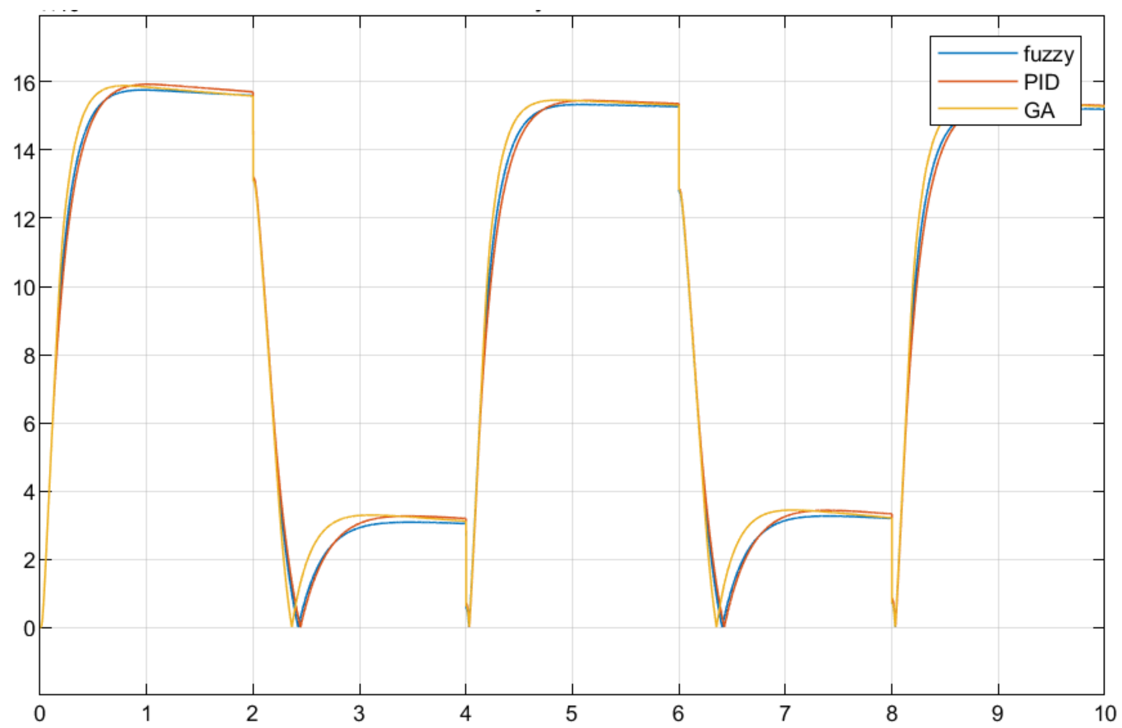


Figure 24. Cost comparison between controllers in Simulink

In the figure 25 can be seen that there is not much variation between different controllers in Simulink. That is why in rest of the results is presented performance of all different controllers in both Simulink and Mevea systems.

4.2 Model comparison

Comparison between Mevea and Simulink models are made by importing input signal that is generated in Simulink to both modelled systems. In figures 25 ,27 and 29 below are presented piston displacement in both system using Simulink generated input signals that are generated using different controlling methods. In figures 26, 28 and 30 is presented IAE of every controlling method using step input at one second from 0.5 to 0.8.

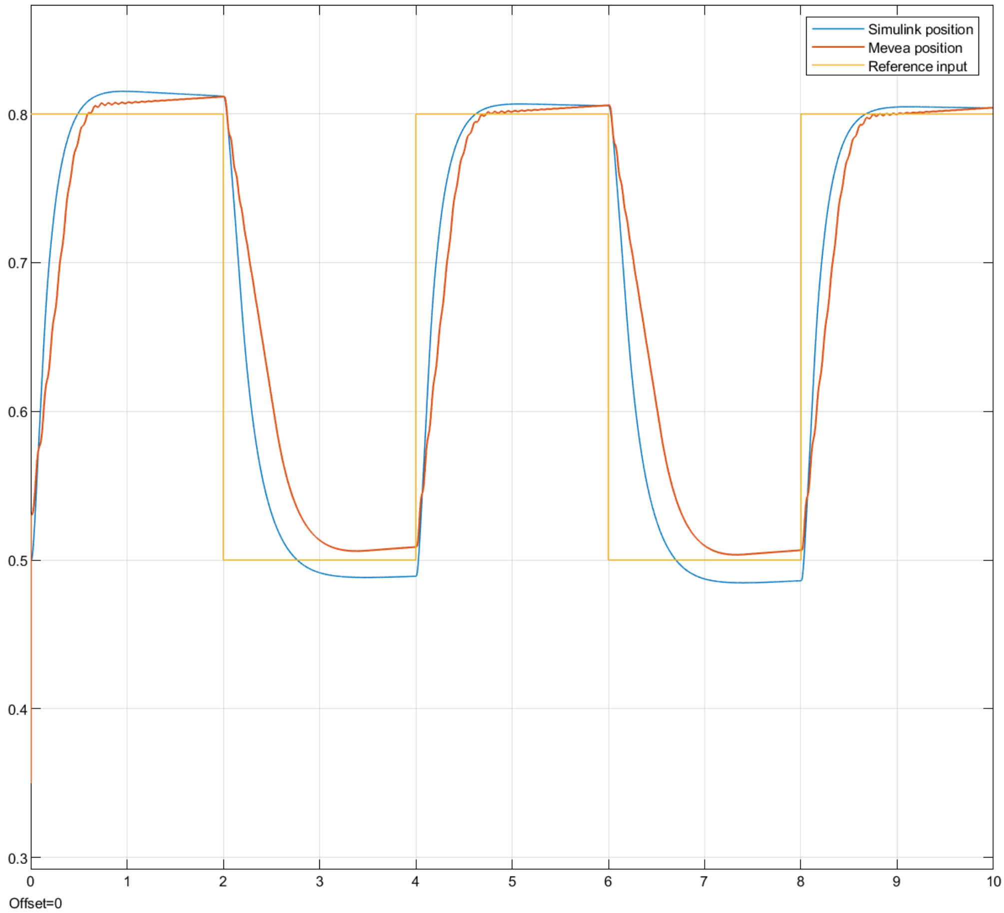


Figure 25. Piston displacement in x-axis using fuzzy logic controller

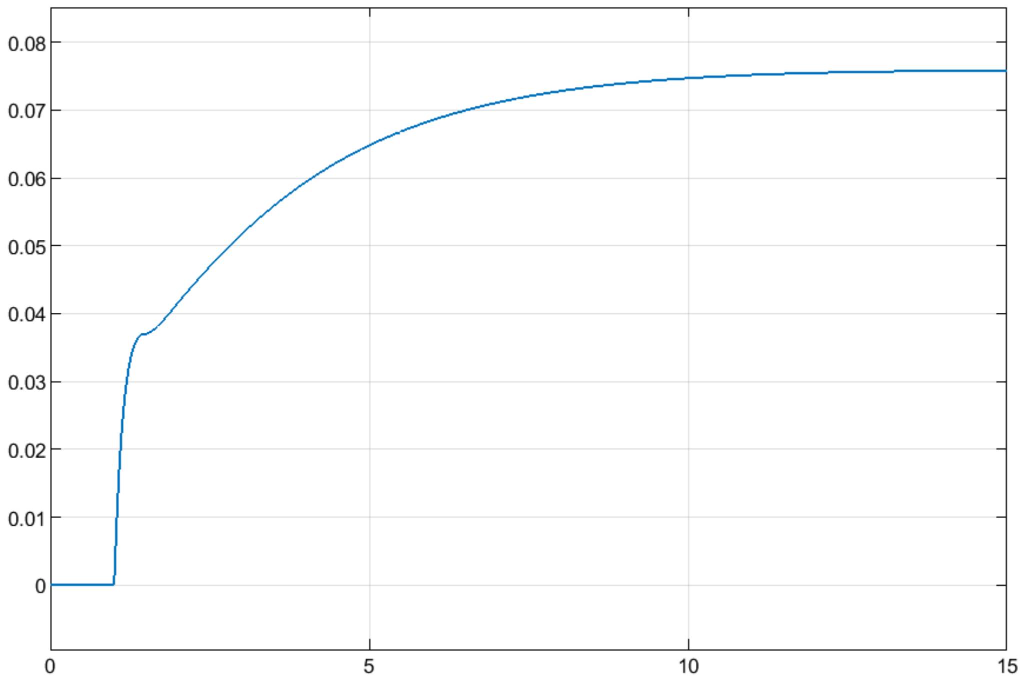


Figure 26. IAE of the system behavior using fuzzy logic controller.

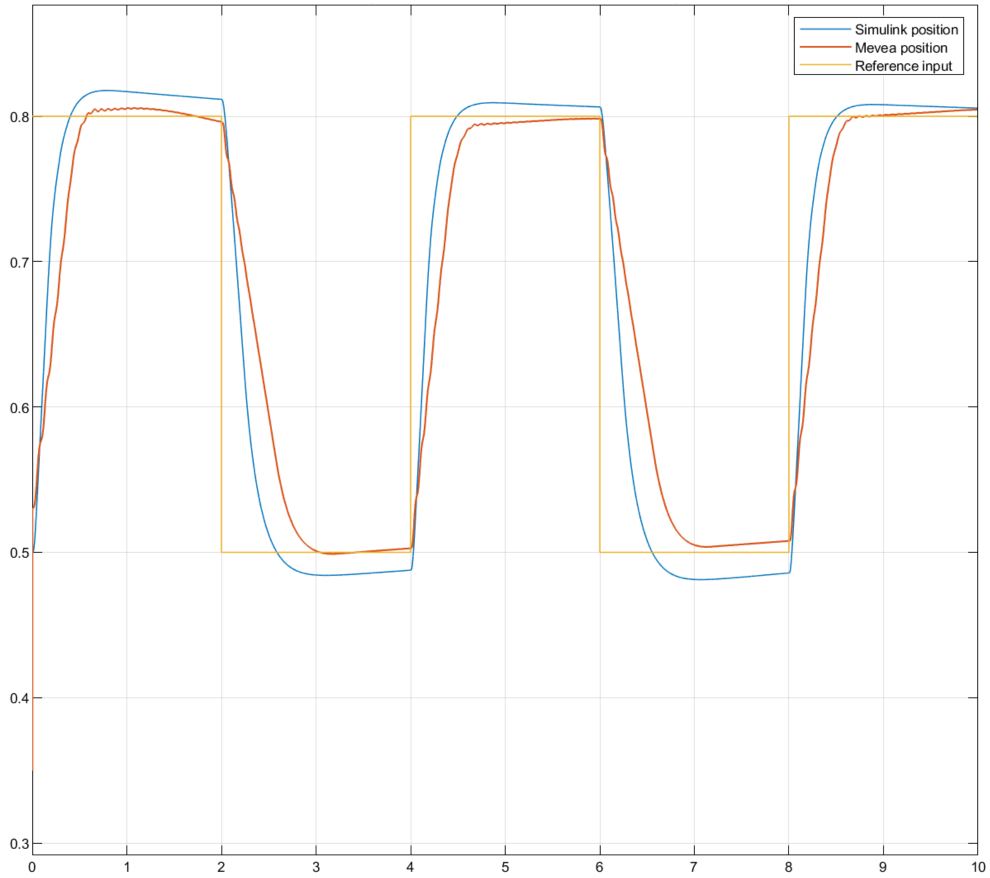


Figure 27. Piston displacement in x-axis using manually tuned PID- controller.

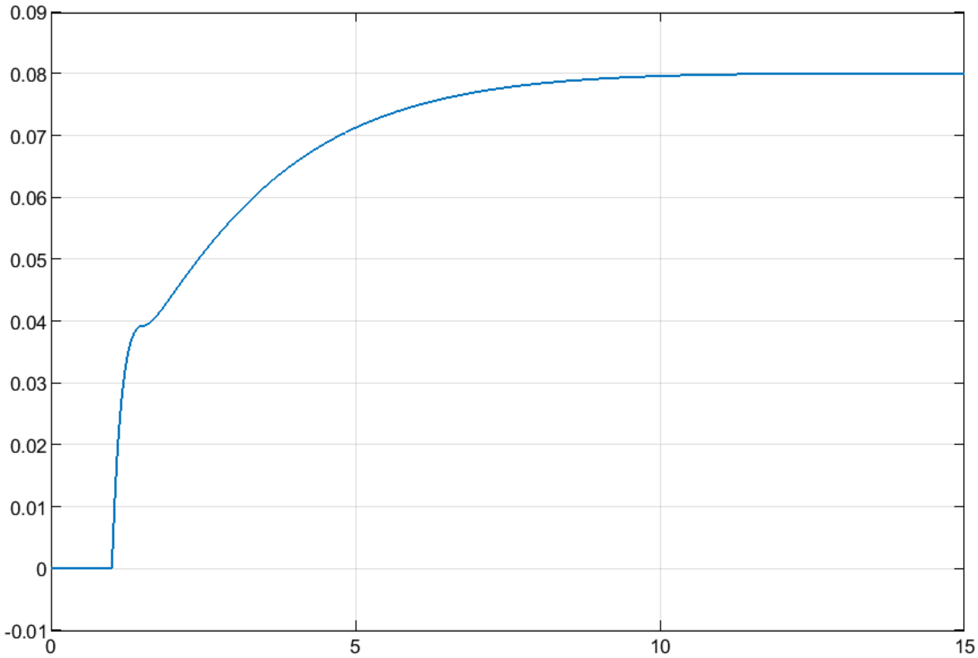


Figure 28. IAE of the system behavior using manually tuned PID-controller.

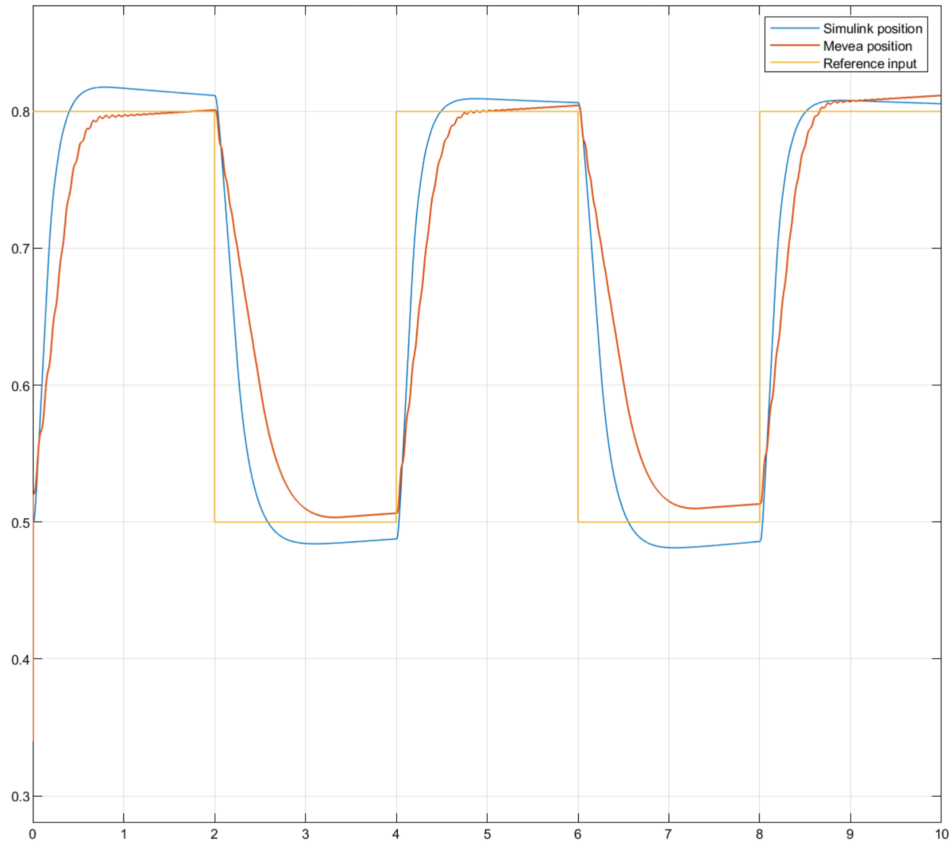


Figure 29. Piston displacement in x-axis using GA tuned PID- controller.

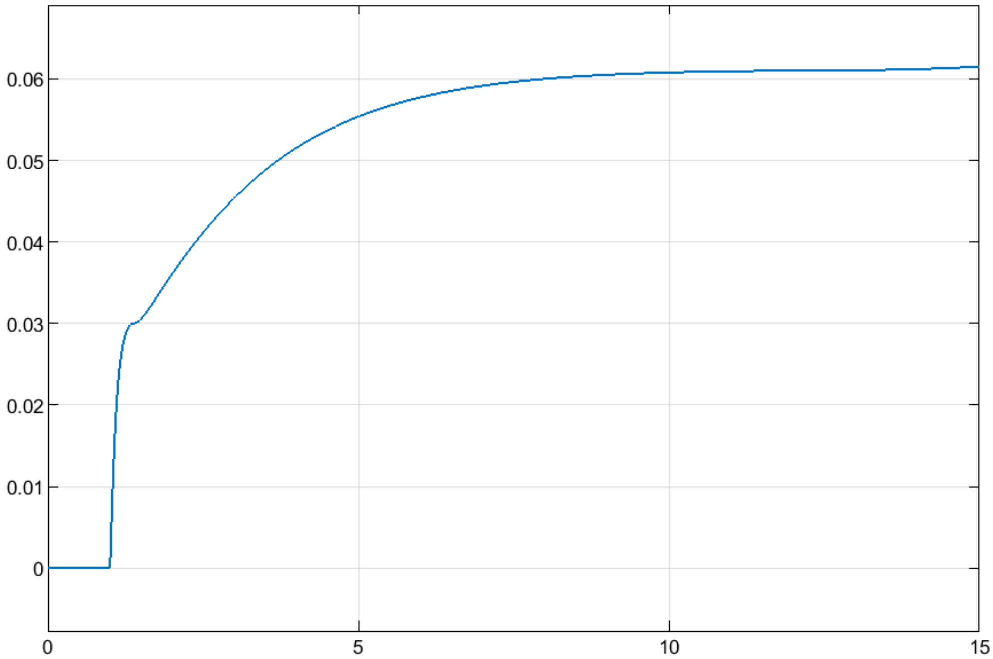


Figure 30. IAE of the system behavior using GA tuned PID-controller.

Where IEA rises above 7 with fuzzy logic controller and to 8 when using manually tuned PID- controller, provides the GA tuned PID- controller best results in IAE comparison. GA tuning script expects the reference input to be step input, which makes its behavior best to correspond with step input.

From these figures can be clearly seen that in Mevea model the responses rise time from 0.5 meters to 0.8 meters is smaller than it is from 0.8 meters to 0.5 meters. This may be caused by unproperly modelled hydraulics in Mevea that makes the system behave unbalanced. Rise time is relatively large is both systems and it could be neglected by adding an amplifier to the system after the controller. In the figure 31 below, can be seen the effect of amplifier to systems behavior.

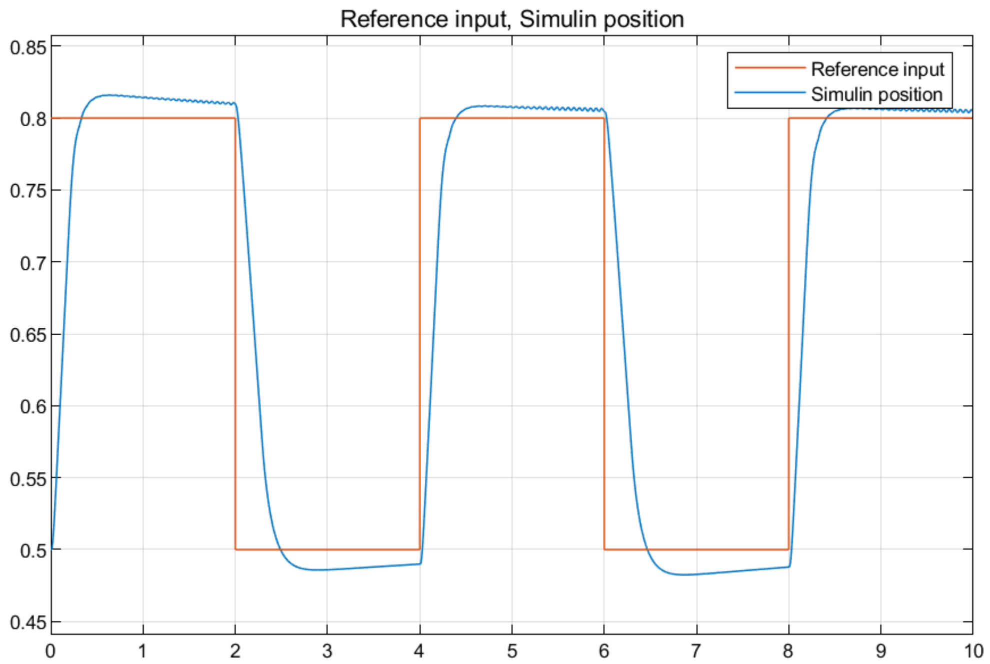


Figure 31. Adding amplifier to system.

Rise time of the Simulink position decreased from 0.4 to 0.3 with a simple amplifier, which multiplied the input signal by two. Overshoot and other characteristics of the position output remained the same after adding the amplifier to the system.

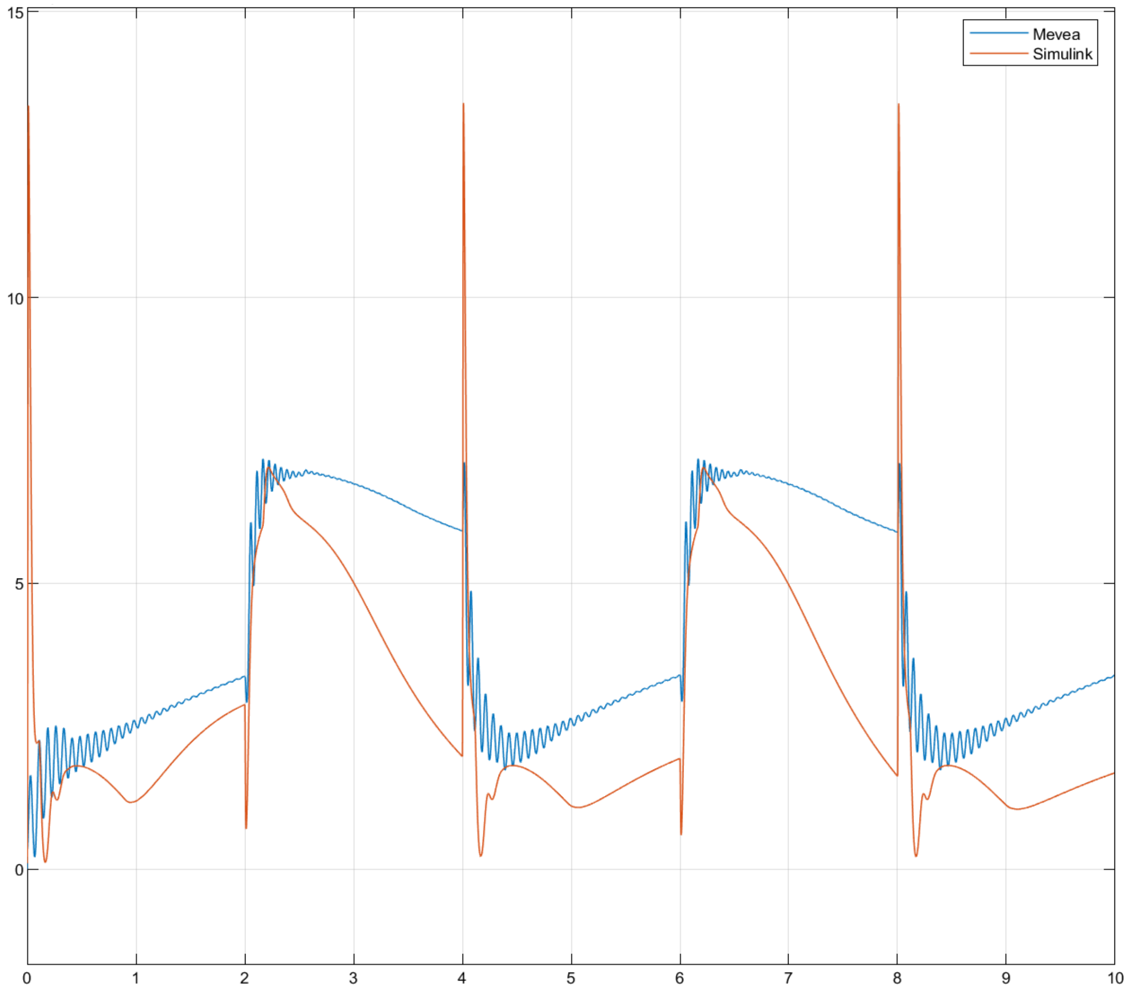


Figure 32. Cylinder side pressures [MPa], signal generated by Simulink manually tuned PID- controller.

Simulink model cylinder side pressure behavior could be caused by unsaturated system, where the input voltage does not reach its maximum or minimum. By taking a look to figures 33 and 33 it is clear that both control voltage and valve input voltage reaches its maximum and minimum, so Simulink system behavior is caused by using the simplified leakage model. The peak in pressures, both in Mevea and Simulink model may be caused by the peaks in valve controlling voltage, which can be seen in figure 34. Another possible reason for peaks in pressures is valve's response, which is not fast enough when moving from negative to positive input signal. The problem with valve response when moving from negative to positive signal can clearly be seen at the time of 4 and 8 seconds in simulation. In upper position Simulink drops pressure faster, when compared to Mevea. The effect of this can be seen in figures 26, 28 and 30 as well, where Simulink model fails to hold the position still at desired level. Where Mevea model holds the position much better.

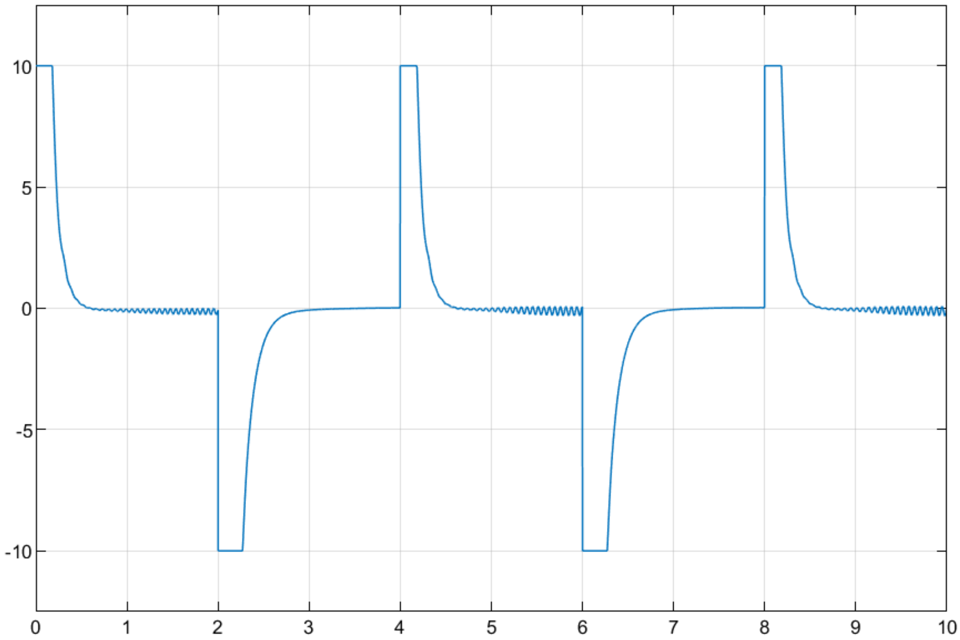


Figure 33. Controlling input from controllers in Simulink, using manually tuned PID-controller.

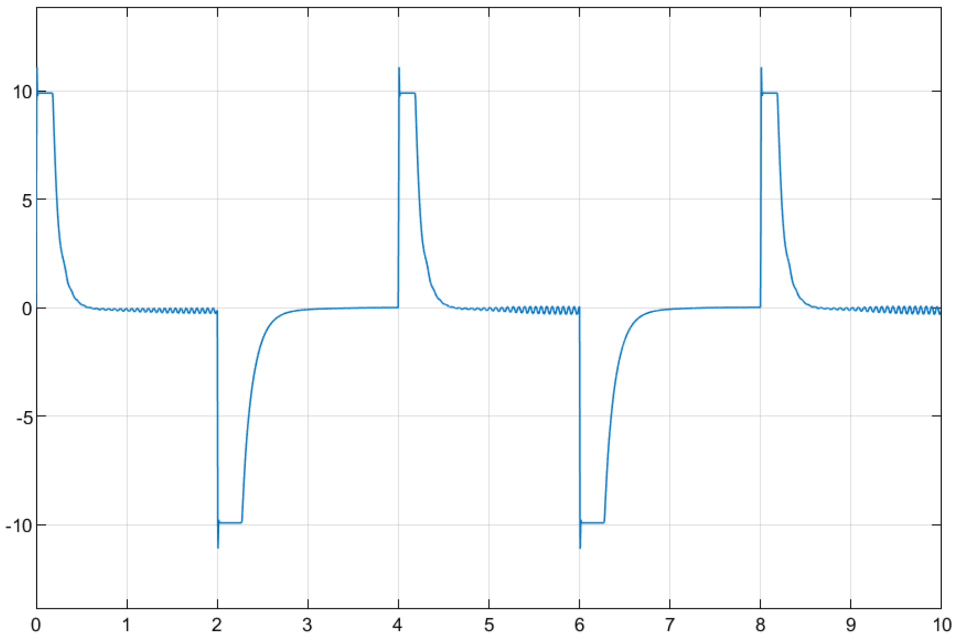


Figure 34. Formed valve controlling voltage in Simulink, using manually tuned PID-controller.

Peaks in formed valve controlling voltage may be caused when taking a derivative from voltage in figure 33 as shown in formula 4. This is only mathematical problem and it could be neglected by adding saturation from -10 to +10 to Simulink model. This could provide much better response in both cylinder and piston side pressures as well.

5 DISCUSSION

This chapter considers different aspect, for example: how useful this research is, how reliable are the formed models and is it possible and useful to continue this research even further. The research is scoped into so tight area, so discussion is not only limited to this research.

5.1 Model comparison

Formed Simulink model have been tested alongside with the real hydraulic servo system in previous researches and Mevea model behaves in the same way as the Simulink model does. Simulink model that was formed in this research differs from other researchers Simulink model in the way that in this research leakage model is simplified. Therefore, it is safe to say that both mathematical models' mimics behavior of the hydraulic servo system. These formed mathematical models does not provide any information for other application by themselves but used theorems and approaches may be useful for similar researchers. Formed Mevea model did not lean that much on the model that was formed in Simulink. Only some initial values, for example like initial pressures and dimensions were applied to Mevea model. Where Simulink model was formed by using know formulas and theorems, where Mevea model was only formed by using known hydraulic and model components and dimensions. Even Mevea model is easier to form it provides more information about the surveyed system than Simulink does, due to multiple functions that have been built in Mevea.

5.2 Controlling methods comparison

Where manually tuned PID- controller is most used controlling method in industry applications, are GA based tuning and fuzzy logic more rarely used. Where all three different controlling methods provides similar response, does the usability factor pay a major role when selecting suitable controlling method. Obviously manually tuned PID- controller provides the controlling method for the process most easily. In GA and fuzzy logic, it is mandatory to determine plants transfer function, where PID- controller manual tuning requires just testing different values for PID- controller. Comparison of GA and fuzzy logic is more difficult. GA tuned PID can provide better response than manually tuned PID-controller, but it requires coding and understanding how GA works. Selection of suitable fitness function, error function and determining transfer function takes a lot time and might include errors. Fuzzy logic controller parameters can be determined in many ways, but the main goal of fuzzy logic parameters to is set setpoints for membership function and rules, which finally forms the brains of fuzzy logic controller. When controlling more complex system by fuzzy logic, it is possible that the amount of membership functions and rules grows too high.

5.3 Further researches and model improvements

Used controlling methods can be implemented to other Simulink or Mevea models as well, but not in real time. Even though the idea was to form completely real-time platform for controlling the models, fuzzy logic controller and GA tuned PID- controller optimization does not run real-time. Further researches could be considering forming completely real-time platform, where the model can be just dragged and dropped to the platform and by determining desired reference input, the platform would form optimal parameters for controllers. In that research it would not be necessary to build any existing models, just finding the right connections between MATLAB's optimization toolboxes, functions, and Simulink models. Simplified concept from this further research is shown in figure 35.

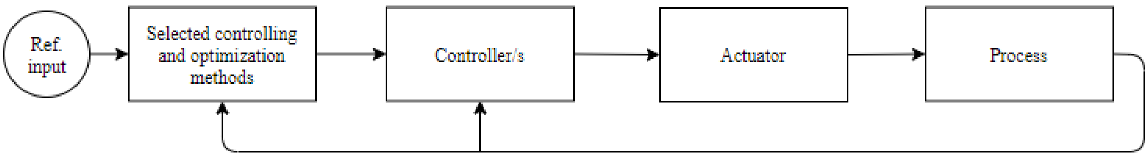


Figure 35. Further research concept.

Controlling and optimization methods would take process error signal and reference input as input and then feed controlling parameters to controllers. This connection between controllers and optimization methods would not be direct, parameter optimization loop would just feed parameters to controllers. Controllers would take normal inputs, reference input and error signal from the process. Using this kind of concept, it would be possible to test any virtually modelled system and optimize its behavior by determining desired movement of the system.

6 CONCLUSION

The initial goal of the thesis is presented in figure 1 in introduction chapter. Testing the real system was not possible in this research due to the problems that occurred during the thesis, so Mevea model was tested as the real system in this research. By taking off the real system testing, final concept was created, which is shown in figure 23. In the introduction chapter goals for this thesis was presented in task list, which was following:

1. Conduct literature review from the topic.
2. Develop Simulink based model of the hydraulic servo system.
3. Develop Mevea based model of the hydraulic servo system.
4. Test both models with manually tuned PID- controller.
5. Develop different controlling and optimization methods.
6. Test the real system in real-time using different controllers and input signals that are created in Simulink or Mevea.

Could be said that this research provides enough information for repeating parts 1-5. Unfortunately, it was not possible to feed formed input signals to real system, but it did not play that major role in this thesis. By combining formed models to real system, it would provide couple more figures to results that were already obtained. It would be possible to optimize and test more widely with the connection to real system. Some behaviors are neglected from formed mathematical based models and therefore they do not provide as accurate results as real system would.

Development of both mathematical models are shown in chapter 3, where modelling processes are presented step by step. In same chapter 3 is shown development of different controllers in Simulink and how mathematical models are combined with each other. Obtained results shows that both models work correctly with different controllers and have the same behavior using same controlling input.

One goal was to create universal platform for controlling different processes, where ideally any mathematically modelled process could replace models that are modelled in this research. This was not tested in this research, but all the controlling methods that were used can be easily modified to fit any system. So unfortunately, it is not possible to say that task 6 was completed in this research. By conducting research using the idea that is presented discussion chapter in figure 29, would it be possible to obtain more valuable results in scientific matter.

In the end can be said that the results of this research can be seen as positive. Both of the mathematical models that were formed works well with selected controllers. The results and discussion in this research provide excellent ideas for further research, where the findings in this research could work as trendsetter for further researches or ideas.

LIST OF REFERENCES

Almabrok A., Psarakis M. & Dounis A. 2018. Fast tuning of the PID controller in An HVAC System Using the Big Bang-Big Crunch Algorithm and FPGA Technology. Basel: MPDI Journals. P. 7.

Alfaro V. M. & Vilanova R. 2016, Model-Reference Robust Tuning of PID Controllers. New York: Springer International Publishing. P. 1–3.

Ansari A. Q. 1998. The Basics of Fuzzy logic: A Tutorial Review. Stafford: Computer Education Group.

Belanger J., Venne. P & Paquin J-N. 2010. The what, where, and why of real-time simulation. Montreal: Opal-RT Technologies. P. 37–42.

Bosch Rexroth AG. 2019. Catalog for Industrial Hydraulics. [Referred 23.1.2020] Available at: https://www.transpoint.ru/upload/iblock/ea0/ea07c87511d68bb7942654d499bd2e6c.pdf

Cheng Y., Lu C. & Ma J. 2015. Health Assessment for hydraulic servo system using manifold learning based on EMD. Transactions of the Canadian Society for Mechanical Engineering. DOI: http://dx.doi.org/10.1139/tcsme-2015-0045.

Comer E. R. 2005. Methods and systems for providing simulation-based technical training. [Referred 13.4.2020]. Available at: http://www.freepatentsonline.com/y2005/0095572.html

Gurunatham. S. K. 2003. Interactive Shape Optimization for Aesthetics using Genetic Algorihms. In: EDS PLM User Conference. Pune, India. P. 1–3, 15–16 & 56. Available at: https://www.researchgate.net/publication/296057976_Interactive_Shape_Optimization_for_Aesthetics_using_Genetic_Algorithms

Kulakowski B. T., Gardner. J. F & Shearer J. L. 2007. Dynamic modeling and control of engineering systems. Cambridge: Cambridge University Press. P. 230-233.

Mathworks, 2020. Simulation and Model-Based Design. [Referred 22.4.2020]. Available at: https://www.mathworks.com/products/simulink.html

MEVEA. 2020a. Beginner tutorials. Lappeenranta: Mevea ltd.

MEVEA. 2020b. The Digital Twin at the center of R&D. Lappeenranta: Mevea ltd.

Murray R. M. 2006. Lecture: Linear Quadratic Regulator. California Institute Of Technology.

Nuruzzaman M. 2014. Control system analysis & design in MATLAB and Simulink. Raleigh: LULU Press, Inc. P. 110-109)

Plummer A. R. & Vaughan N. D. 1996. Robust adaptive control for hydraulic servosystems. Robust adaptive control for hydraulic servosystems. ASME, Journal of Dynamic Systems, Measurement and Control. DOI: http://dx.doi.org/ 10.1115/1.2802309.

Parr E. A. 2003, Programmable Controllers – An engineer's guide, third edition. Reed Elsevier plc group. P. 1–4.

Popovici. K & Mosterman. P. J. 2017. Real-time Simulation technologies: Principles, Methodologies and Applications. New York: CRC Press. P. 1-2.

Qi Q., Tao F., Zuo Y. & Zhao D. 2019. Digital Twin Service towards Smart Manufacturing. Elsevier Inc. P. 237–238.

Rabie M. G. 2009. Fluid Power Engineering. New York: The McGraw-Hill Companies, Inc. P. 201

Reeves C. R. 2010. Genetic Algorithms. New York: Springer International Publishing. P. 56–57.

Roozbahani H., Handroos H & Wu H. 2007. Robust adaptive control of a hydraulic servo system by utilizing real-time simulation.

Yousefi H, 2007. On Modelling, System Identification and Control of Servo-Systems with a Flexible Load.

Sivanandam S. N & Deepa S. N. 2008, Introduction to Genetic Algoritms.
New York: Springer-Verlag Berlin Heidenberg. p. 1–3 & 15–16.

Totten G. E. 1999. Handbook of Hydraulic Fluid Technology. New York: Marcel Dekker, Inc. P. 257 & 684.

Zhong J. 2006. PID Controller Tuning: A Short Tutorial. [Referred 2.3.2020] P. 1–13. Available at: https://docplayer.net/20872836-Pid-controller-tuning-a-short-tutorial.html

Visioli. A. 2006. Practical PID control. London: Springer-Verlag London Limited. P. 2–3.

Sivanandam S. N, Sumathi S & Deepa S. N. 2007. Introduction to Fuzzy Logic using MATLAB. Berlin: Springer-Verlag Berlin Heidelberg. P. 73-74, 77 & 113-115.

Liu Y. F., Li J., Zhand Z. M., Hu X. M & Zhang W.J. 2015. Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system. DOI: https://doi.org/10.5194/ms-6-15-2015

Lewotsky, K. 2015. Servo Pumps Give Hydraulics End-Users An Edge. [Referred 17.3.2020]. Available at: https://www.motioncontrolonline.org/content-detail.cfm/Motion-Control-Technical-Features/Servo-Pumps-Give-Hydraulics-End-Users-An-Edge/content_id/

APPENDIX

Appendix I. Boss Rexroth AG Type 4WRPEH 6 valve catalog.

Technical data

General

Construction		Spool type valve, operated directly, with steel sleeve
Actuation		Proportional solenoid with position control, OBE
Type of mounting		Subplate, mounting hole configuration NG6 (ISO 4401-03-02-0-94)
Installation position		Optional
Ambient temperature range	°C	–20 … +50
Weight	kg	2.7
Vibration resistance, test condition		max. 25 *g*, shaken in 3 dimensions (24 h)

Hydraulic (measured with HLP 46, $\vartheta_{oil} = 40\,°C \pm 5\,°C$)

Pressure fluid		Hydraulic oil to DIN 51524 … 535, other fluids after prior consultation					
Viscosity range – recommended	mm^2/s	20 … 100					
Viscosity range – max. permitted	mm^2/s	10 … 800					
Pressure fluid temperature range	°C	–20 … +70					
Maximum permissible degree of contamination of pressure fluid Purity class to ISO 4406 (c)		Class 18/16/13 [1]					
Flow direction		See symbol					
Nominal flow at $\Delta p = 35$ bar per notch [2]	l/min	2	4	12	15	24	40
Max. working pressure	bar	Port P, A, B: 315					
Max. pressure	bar	Port T: 250					
Operating limits at Δp Pressure drop at valve $q_{Vnom}: > q_N$ valves (symbol 1)	bar	315	315	315	315	315	160
(symbol 2)	bar	315	315	315	280	250	100
Leakage at 100 bar (symbol 1)	cm^3/min	<150	<180	<300	–	<500	<900
(symbol 2)	cm^3/min	–	–	–	<180	<300	<450

Static/Dynamic

Hysteresis	%	≤ 0.2
Manufacturing tolerance for q_{max}	%	< 10
Responce time for signal change 0 … 100 %	ms	≤ 10
Thermal drift		Zero point displacement < 1 % at $\Delta T = 40\,°C$
Zero adjustment		Factory-set ±1 %
Conformity		CE EN 61000-6-2, EN 61000-6-3

[1] The purity classes stated for the components must be complied with in hydraulic systems. Effective filtration prevents problems and also extends the service life of components. For a selection of filters, see catalogue sections RE 50070, RE 50076 and RE 50081.

Appendix II. Part properties for MEVEA.

Mass sledge			
	x	**y**	**z**
Position [m]	0,6	1,8	0
Center of mass [m]	0,496	0,073	0,3
Mass [kg]	210		
Body Type	Ridig		
Relative to body	Slide		
Inertia frame	Center of Mass		
Moments and products of Inertia [kg*m^2]			
	x	**y**	**z**
x	4,986	0	0
y	0	16	0
z	0	0	16

Piston			
	x	**y**	**z**
Position [m]	0	0	0
Center of mass [m]	0,591	0	0
Mass [kg]	5		
Dummy type	B2BF		
Relative to	Translational force		
Inertia frame	Center of Mass		
Moments and products of Inertia [kg*m^2]			
	x	**y**	**z**
x	0	0	0
y	0	1,08	0
z	0	0	1,08

Slide			
	x	**y**	**z**
Position [m]	0	0	0,3
Center of mass [m]	0,804	0,118	0,3
Mass [kg]	370		
Body Type	Ridig		
Relative to body	Ground		
Inertia frame	Center of Mass		
Moments and products of Inertia [kg*m^2]			
	x	**y**	**z**
x	26,8	-7,2	0
y	0	189,9	0
z	0	0	168,6

Cylinder			
	x	**y**	**z**
Position [m]	0,6	1,8	0
Center of mass [m]	0,496	0,073	0,3
Mass [kg]	10		
Dummy type	B2BF		
Relative to	Translational force		
Inertia frame	Center of Mass		
Moments and products of Inertia [kg*m^2]			
	x	**y**	**z**
x	0	0	0
y	0	5	0
z	0	0	5

Name	Ground.slide		
	x	**y**	**z**
Position [m]	0	0	0
Orientation [rad]	-pi/2	0	0

Name	Slide.ground		
	x	**y**	**z**
Position [m]	0	0	0.3
Orientation [rad]	-pi/2	0	0

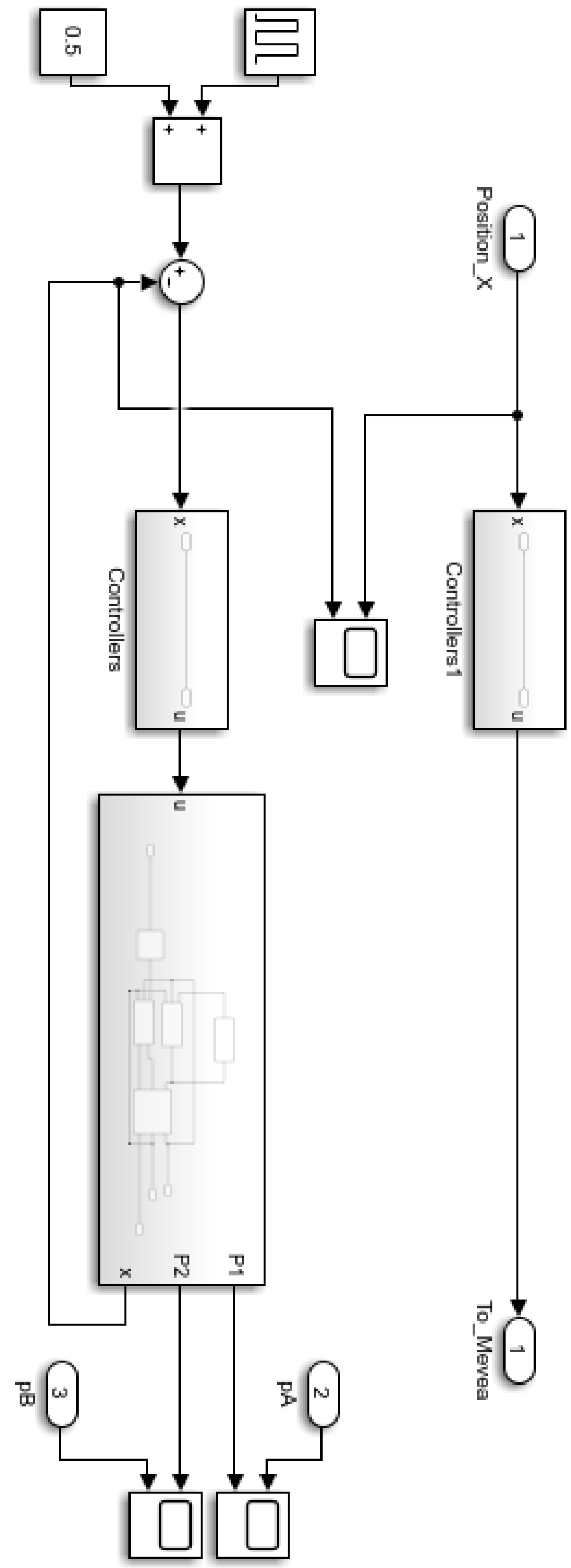
Name	Slide.sledge		
	x	**y**	**z**
Position [m]	-0.5	0	0.3
Orientation [rad]	pi/2	pi/2	0

Name	Sledge.slide		
	x	**y**	**z**
Position [m]	0.2	0.18	0.3
Orientation [rad]	pi/2	pi/2	0

Name	Slide.force		
	x	**y**	**z**
Position [m]	0	0.3	0.3
Orientation [rad]	0	0	0

Name	Sledge.force		
	x	**y**	**z**
Position [m]	-0.3	-0.12	0.3
Orientation [rad]	0	0	0

Appendix IV. Modelled system in Simulink.



Appendix V. MATLAB script for GA optimization, Sharma 2017.

```
function[J] = pid_optimi(x)

s = tf('s')

plant = ... %Transfer function of the plant

Kp = x(1)
Ki = x(2)
Kd = x(3)

cont = Kp + Ki/s Kd * s; %Pid controller

dt = 0.01; %Time step
t = 0:dt:4; %

%step(feedback(plant*cont,1)) %For plotting the step response
e = 1 - step(feedback(plant*cont,1),t); %Error

J = sum(t'.*abs(e)*dt); %Fitness function
```

Appendix VI. Simulink external interface.