

LUT UNIVERSITY
LUT School of Energy Systems
LUT Mechanical Engineering

Jomi Kotta

**MODELING OF PATU-655 LOADER WITH A COMBINATION OF MEVEA AND
SIMULINK USING FUNCTIONAL MOCK-UP INTERFACE**

22.11.2021

Examiners: Professor Heikki Handroos
D. Sc. (Tech.) Victor Zhidchenko

TIIVISTELMÄ

LUT-Yliopisto
LUT School of Energy Systems
LUT Kone

Jomi Kotta

Patu-655 kuormaimen mallinnus Mevean ja Simulinkin yhdistelmällä käyttäen Functional Mock-up Interface rajapintaa

Diplomityö

2021

54 sivua, 34 kuvaa, 3 taulukkoa ja 1 liite

Tarkastajat: Professori Heikki Handroos
Tkt Victor Zhidchenko

Hakusanat: yhteissimulaatio, systeemisimulointi, monikappalesysteemi, reaaliaikainen simulaatio

Tämän diplomityön tavoitteena oli luoda Patu-655 kuormaimelle simulaatiomallit Mevealla (mekaniikka) ja Simulinkillä (hydrauliikka) ja yhdistää mallit Functional Mock-up Interface (FMI) -rajapinnan avulla. Mekaniikkamalli Meveassa koostuu pilarista, nostopuomista, 4-tankomekanismista, kallistuspuomista, jatkopuomista ja kourasta. Simulinkin hydraulipiirissä on kaksi hydraulisylinteriä, kaksi proportionaalista suuntaventtiiliä ja kuormituksen tunnistava pumppu. Simulaatiomallit yhdistettiin käyttämällä FMI for Co-Simulation with generated code -menetelmää. Aiemmin Mevea ja Simulink on yhdistetty käyttämällä tool coupling -menetelmää, joka voi tehdä simuloinnista laskennallisesti raskasta ja voi vaatia suuremman aika-askelen toimiakseen oikein.

Tutkimus alkoi tekemällä hydraulipiiri Simulinkillä ja mekaniikkamalli Mevealla. Mallit testattiin yksittäin, jotta ne voitiin varmistaa toimivaksi ilman, että ne vaikuttivat toisiinsa. Hydraulipiiri testattiin liittämällä se aiemmin valmistettuun Simscape Multibody malliin ja Mevean mekaniikkamalli testattiin rakentamalla hydraulipiiri Mevean sisäänrakennetulla työkalulla. Kun molemmat mallit todettiin toimiviksi, ne yhdistettiin FMI:n avulla, jossa Mevea toimi master ohjelmistona ja Simulinkin malli slave ohjelmistona.

Simulaatiomallia verrattiin laboratoriomittausten tuloksiin ja aiemmin Simscape Multibodylla kehitettyyn malliin. Tässä diplomityössä kehitetty malli saavutti samanlaisia tuloksia kuin aiemmat tutkimukset ja mittaukset paineiden, sylinterivoimien ja asemien suhteen, mutta oli simulointiajan suhteen huomattavasti nopeampi kuin aiempi malli, koska tämä malli pystyi suorittamaan simulaation reaaliajassa, kun mekaniikkamallissa oli 1 ms aika-askel ja 0.25 ms hydrauliikassa. Tämän diplomityön tulosten perusteella FMI:n käyttö on sopiva vaihtoehto sellaisten järjestelmien simulointiin, jotka voivat hyötyä useiden simulointiohjelmistojen samanaikaisesta käytöstä, sillä FMI:n käyttö ei kasvattanut simulointiaikaa ollenkaan tässä mallissa.

ABSTRACT

LUT University
LUT School of Energy Systems
LUT Mechanical Engineering

Jomi Kotta

Modeling of Patu-655 loader with a combination of Mevea and Simulink using Functional Mock-up Interface

Master's thesis

2021

54 pages, 34 figures, 3 tables and 1 appendix

Examiners: Professor Heikki Handroos
D. Sc. (Tech.) Victor Zhidchenko

Keywords: co-simulation, system simulation, FMI, multibody system, real-time simulation

The aim of this master's thesis was to create a simulation models for Patu-655 loader with Mevea (mechanics) and Simulink (hydraulics) and then combine the models with Functional Mock-up Interface (FMI). The mechanics model in Mevea consists of pillar, lift boom, 4-bar mechanism, tilt boom, extension boom and log grapple. The hydraulic circuit in Simulink has two hydraulic cylinders, two proportional directional control valves and a load sensing pump. The combination of the simulation models was achieved by using FMI for Co-Simulation with generated code method. Previously the interaction between Mevea and Simulink has been achieved with tool coupling method which can make the simulation computationally heavy and may require higher timestep to work correctly.

The research started by building hydraulic circuit with Simulink and multibody model with Mevea. Both models were tested independently, so that they could be verified without affecting each other. Hydraulic circuit was tested by connecting it to previously developed Simscape Multibody model and Mevea was verified by building a hydraulic circuit with Mevea's in-built tool. After both systems were verified as working, they were connected with FMI, where Mevea acts as master software and Simulink as slave software.

The simulation model was compared to data from laboratory measurements and previously developed simulation model which was developed with Simscape Multibody and Simulink. The model developed in this thesis achieved similar results to previous studies and laboratory measurements in terms of pressures, cylinder forces and positions, but was significantly faster than previous model, as it was able run the simulation in real time with time step of 1 ms for mechanics and 0.25 ms for hydraulics. Based on the results from this thesis, using FMI is a valid option for simulating systems which can benefit from using multiple different simulation software as the use of FMI didn't increase the simulation time at all in this model.

TABLE OF CONTENTS

TIIVISTELMÄ

ABSTRACT

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

1	INTRODUCTION	6
	1.1 Motivation of the research	8
	1.2 Goals	9
	1.2.1 research questions	9
	1.2.2 Hypothesis	10
	1.3 Research methods	10
2	METHODS.....	11
	2.1 System under investigation.....	11
	2.2 Mechanics model in Mevea	12
	2.3 Hydraulics model in Simulink	17
	2.3.1 Directional control valve equations	17
	2.3.2 Load sensing pump equations.....	18
	2.3.3 Hydraulic cylinder equations.....	19
	2.4 Functional Mock-up Unit.....	21
3	RESULTS	27
	3.1 Lift cylinder simulation	28
	3.2 Tilt cylinder simulation.....	32
4	DISCUSSION.....	37
	4.1 Comparison to previous research.....	37
	4.2 FMU hydraulics compared to Mevea’s hydraulics.....	42
	4.3 Key findings.....	48
	4.4 Topics for future research	48
5	CONCLUSION	49
	LIST OF REFERENCES.....	52
	APPENDIX	

Appendix I: Additional simulation results

LIST OF SYMBOLS AND ABBREVIATIONS

A	Area [m ²]
B	Bulk modulus [Pa]
b	Viscose friction coefficient [Ns/m]
C_v	Semi-empirical flow rate coefficient [-]
F	Force [N]
F_μ	Friction force [N]
F_c	Coloumb friction [N]
K_p	Pump semi-empirical constant [-]
p	Pressure [Pa]
Q	Volume flow [m ³ /s]
U	Voltage [V]
v	Velocity [m/s]
V	Volume [m ³]
v_c	Coloumb velocity threshold [m/s]
Δp_{ref}	Reference pressure drop of the load-sensing pump [Pa]
τ	Time constant [s ⁻¹]
API	Application Programming Interface
CAD	Computer-aided Design
DCV	Directional control valves
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit

1 INTRODUCTION

Nowadays simulations are important part of product development and research because modern simulation tools allow system's behavior to be studied without building the actual physical prototype, but it also allows monitoring of already existing systems with digital twins. Simulations of mechanical systems reduces the product development cycles, costs with the testing phase, validation and overall makes the re-designing faster. Modern simulation systems are capable of simulating complex mechanical systems in real-time, but currently there is a need for coupling mechanical systems to other subsystems. These other subsystems can be for example hydraulics and electronics, and these can require different solvers and time steps than the mechanical model due to different properties. (Peiret et al. 2018, p. 1-2.) In additional to that, complex systems that consists of multiple subsystems are usually designed by multiple teams and each teams has its own areas and tools. Complex designing projects would also benefit from coupling method because the subsystems should be connected as soon as possible because the integration gets less optimal in the later stages of the process. (Gomes et al. 2018, p. 2.)

One option for building multiphysics simulation models is called co-simulation. Co-simulation allows subsystems to use their own solvers and time steps if needed. The integration of subsystems is achieved by exchanging inputs and outputs at specific communication points and this enables using suitable software for the different subsystems. Most of the time these subsystems can also be executed in parallel which reduced the overall simulation time. (Peiret et al. 2018, p. 2.)

A problem with co-simulation is that the interface between software requires standardized definitions, so that the communication between software is reliable and less laborious to achieve (Peiret et al. 2018, p. 2). Functional Mock-up Interface (FMI) can help with this problem because FMI is a standardized interface which was developed to simplify the communication between different simulation software (Modelica Association 2020, p. 1).

According to Blochwitz et al. "FMI is a tool independent standard for the exchange of dynamic models and for Co-Simulation" and it is used for connecting Functional Mock-up

Units (FMU) to other simulation software (Blochwitz et al. 2012, p. 1). FMU is a simulation model in a ZIP file, which consists of FMI description file in XML format, source code of the model in C language with standardized functions and possible additional FMU data. The description file includes the data used in the FMU and defines the variables in standardized way. The C functions are used for the equations in the model. (Modelica Association 2020 p. 4-9.) Figure 1 introduces the basic principle of how FMU works.

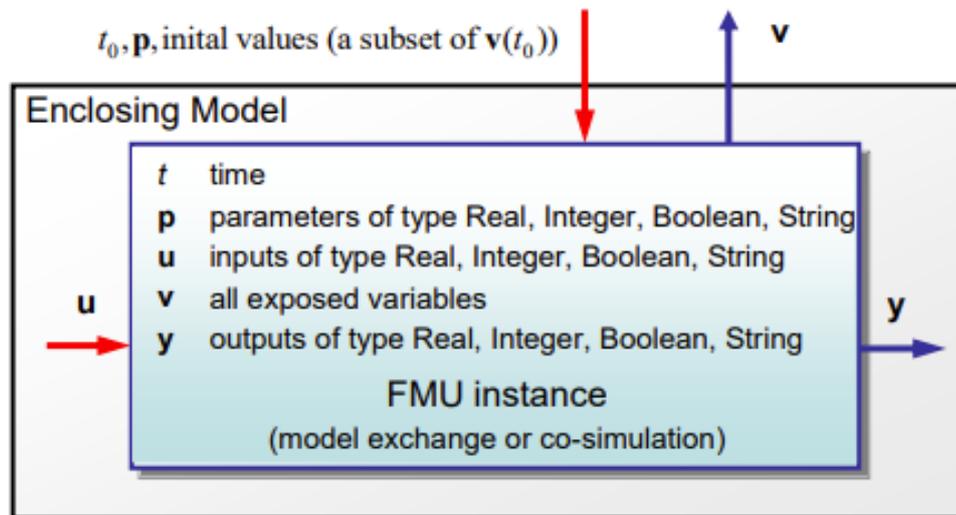


Figure 1. Basic principle of FMU (Modelica Association 2020, p.10).

The enclosing model in figure 1 inputs variables to the FMU, then the FMU interacts with the variables and outputs new values to the enclosing model. For example, in the model built in this thesis, the enclosing model is built with Mevea, which inputs variables to the FMU which is built with Simulink, and the FMU interacts with the variables and then outputs new values to the enclosing model and this interaction happens once a timestep.

FMI has been used in industrial and scientific projects for over a decade, so it isn't completely a new approach but it's getting more popular because it's being developed continuously, and more tools are supporting it. For example, FMI has been used in Mercedes-Benz gearbox projects for software-in-the-loop simulations (Chrisofakis et al. 2011, p. 2), Daimler AG used it for mechatronic shifting simulation (Abel et al. 2012, p. 775) and IFP Energies Nouvelles for combustion engine models (Khaled et al. 2012, p. 453). These are just few examples from first version, FMI 1.0, and nowadays the FMI 2.0 is

supported by over 100 tools and is used by large companies worldwide, such as Robert Bosch GmbH, Haldex, Siemens and Saab AB (Functional Mock-up Interface 2021).

1.1 Motivation of the research

The system under investigation in this thesis is Patu-655 loader, which has been studied and modeled multiple times at LUT University. Previously when systems like Patu-655 have been studied with co-simulation principle using Simulink and Mevea, and Mevea didn't support FMU import and Simulink didn't have option for exporting the model as standalone FMU, the interaction between simulation software has been achieved with Transmission Control Protocol/Internet Protocol socket interface, which can be considered similar to FMI for Co-Simulation with tool coupling method.

FMI for Co-Simulation can be divided into two categories: tool coupling and generated code. The largest difference between these two is that tool coupling requires both modeling software to be open while with generated code only the master program must be running. Using tool coupling can make the simulation heavy for the computer and may require larger timestep to work correctly, especially when the simulation is executed on lower end computer.

Currently, the interaction between Mevea and Simulink has only been studied with tool coupling method and not with generated code option because previously Simulink only supported exporting the model for tool coupling but introduced option for "Export for co-simulation standalone FMU" in MATLAB version R2020a, which is same thing as FMI for Co-Simulation with generated code technique (MathWorks 2021b).

For example, Mäkitalo (2020) studied a forest machine system where mechanics were modeled with Mevea, hydraulics with Amesim and control system with Simulink. In that study Amesim model was exported with FMU for Co-Simulation with generated code, Simulink with tool coupling and Mevea was the master program. Since the generated code option is nowadays supported in Simulink and it hasn't been studied in a system like this, it will be used in this thesis.

Similar approaches for connecting Simulink and Mevea can be found from Khadim (2017) and Ustinov (2018). Both of the studies used Mevea for mechanics and Simulink for controlling. In these studies the Mevea-Simulink interface is achieved with Transmission Control Protocol/Internet Protocol socket interface, which means that Mevea software acts server and Simulink as host.

Although FMI for Co-Simulation with generated code hasn't been studied in LUT University with Simulink and Mevea, it has been used between different software. Kauppila (2019) used it for exporting Amesim model to Adams and connected Adams and Simulink with Adams' Plant export tool which generates .m-file to MATLAB. Gupta (2021) exported motion control algorithm from Simulink to SimulationX with FMI Co-Simulation Target for Simulink®Coder™, which is a third-party add-on for Simulink. FMI Target for Simulink Coder is essentially same thing as MATLAB's own "Export for co-simulation standalone FMU" but it only support FMI 1.0 version (MathWorks 2021a).

In this study Patu-655's mechanics will be modeled with Mevea and hydraulics with Simulink and the models will be connected with FMI for Co-Simulation with generated code method. Shevchuk (2020) modeled the Patu-655 with Simulink and Simscape Multibody in his master's thesis so it will be used for comparing the results because ideally both systems should have similar results even with different software. In that study the system's mechanics were modeled with Simscape Multibody and hydraulic circuit was built with Simulink.

1.2 Goals

The aim of this thesis is to model the Patu-655 loader's mechanics with Mevea and hydraulics with Simulink and then combine these two models with FMI for Co-Simulation with generated code method. The study can be called successful if the FMU version achieves similar results to the Shevchuk (2020) study's simulation results and laboratory measurements while using reasonable timestep and parameter values.

1.2.1 research questions

How does the interaction between Simulink and Mevea work with FMI while using the Patu-655 model?

Is the model developed in this thesis fast enough for real-time simulations or is the model computationally too heavy?

Are the Patu-655 simulation results different with FMU version compared to Simulink-Simscape version and laboratory measurements?

Things to take into consideration while using multiple different software?

What are the benefits of using FMU in this model?

1.2.2 Hypothesis

The hydraulic circuit built in thesis should achieve same results to the Shevchuk (2020) study because both systems model hydraulics with same equations with Simulink, but the mechanics may have differences due to different software. With the FMI for Co-Simulation with generated code, the simulation time should be faster than in previous studies.

1.3 Research methods

The research will start by gathering all the needed equations and parameters for the hydraulic circuit of Patu-655. After that, the hydraulic circuit is built with Simulink and the system can be tested with step input to see how the system responses. The hydraulic circuit can be verified by connecting the circuit to Simscape Multibody model that was used in Shevchuk (2020) study. This way the circuit can be verified without Mevea affecting the results.

Normally the Mevea multibody model building would start by making the 3D models and calculating inertias, but in this thesis Patu-655 3D-models by Luostarinen et al. (2014) were used. So, in this case, the 3D-models can be exported to the Mevea and the multibody model can be built based on the data from Luostarinen et al. (2014). Functionality of the Mevea model can be tested by building hydraulics and control system with Mevea's own tools.

After both models are built, they can be combined by exporting the Simulink hydraulics model to FMU and then it can be imported to Mevea. Then the model can be compared to Shevchuk (2020) simulations and to the laboratory measurements. After the whole system is verified to be working correctly, the FMU hydraulic circuit can also be built with Mevea's hydraulics so that the effects and benefits of FMU can be monitored.

2 METHODS

The research in this thesis can be divided into three main categories which are explained in following chapters. The methods chapter will begin by introducing the system under investigation and continues by explaining how the system's mechanics and hydraulics are built. The last part of methods chapter briefly introduces the basics of FMI and explains how it's being utilized in this thesis.

2.1 System under investigation

The system under investigation is Patu-655 loader from figure 2. It is a hydraulic crane, which has in total five hydraulic cylinders, but in this thesis only the lift and tilt cylinders have been modeled because the other three weren't used. For example, one of the not used cylinders controls the extension boom and it was at the same position during the whole tests, so it was modelled as fixed to the tilt boom. Structurally the loader consists of pillar, lift boom, 4 bar mechanism between lift and tilt boom, tilt boom and extension boom. The 3D models have small differences compared to the real-life system, for example the connecting mechanism between the lift and tilt boom is a bit simplified and instead of a 201 kg mass at the end of the boom, the 3D model has a log grapple which acts as a mass in the simulation, as seen from figure 6.



Figure 2. System under investigation (Shevcuk 2020, p. 28).

2.2 Mechanics model in Mevea

The modeling process of mechanics started by creating bodies and adding graphics for each part. Creating bodies includes adding position and orientation of the body, selecting body type, and defining to which body it's related to, for example the pillar is relative to ground, the lift boom is relative to pillar and the tilt boom is relative to the lift boom. In this model all parts are rigid for simplicity. Because Mevea doesn't support importing 3D models directly from Computer-aided Design (CAD) software, mass properties must be defined manually based on data from the CAD model. 3D models were imported to Mevea in stereolithography format to keep the simulation as light as possible, but the quality of the graphics will be worse. In addition, all unnecessary parts, such as, hoses and screws aren't included in the 3D models to reduce the complexity of graphics. Figure 3 shows all the required body properties for lift boom.

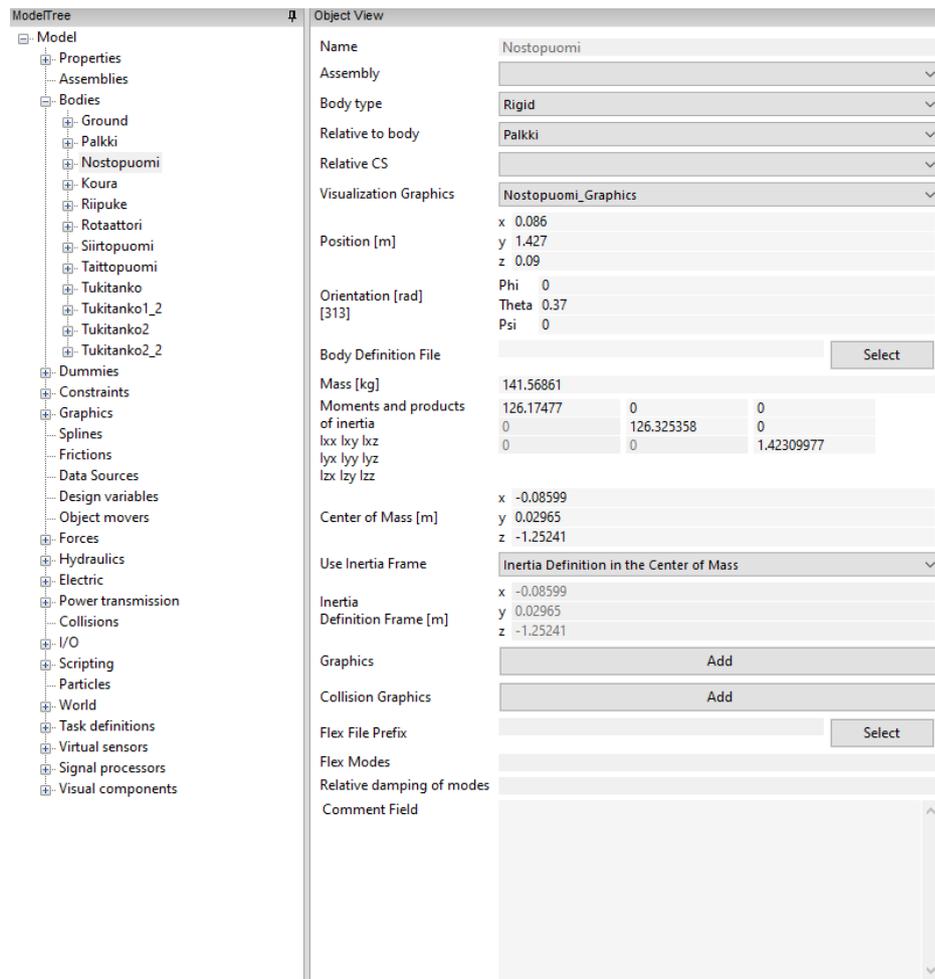


Figure 3. Mevea modeler body properties window.

After the parts are defined and the graphics are imported the next step is to connect the parts by adding constraints. Mevea supports 13 different joint types, but this model only required translational, fixed and revolute joints. As seen from the figure 4, adding a constraint requires selecting the joint type, bodies that are connected to the joint, position of the joint in each body and finally defining joint axis directions in unit vectors u_a , u_b , u_c and u_d . For example in figure 4 the revolute joint rotates around the global y axis.

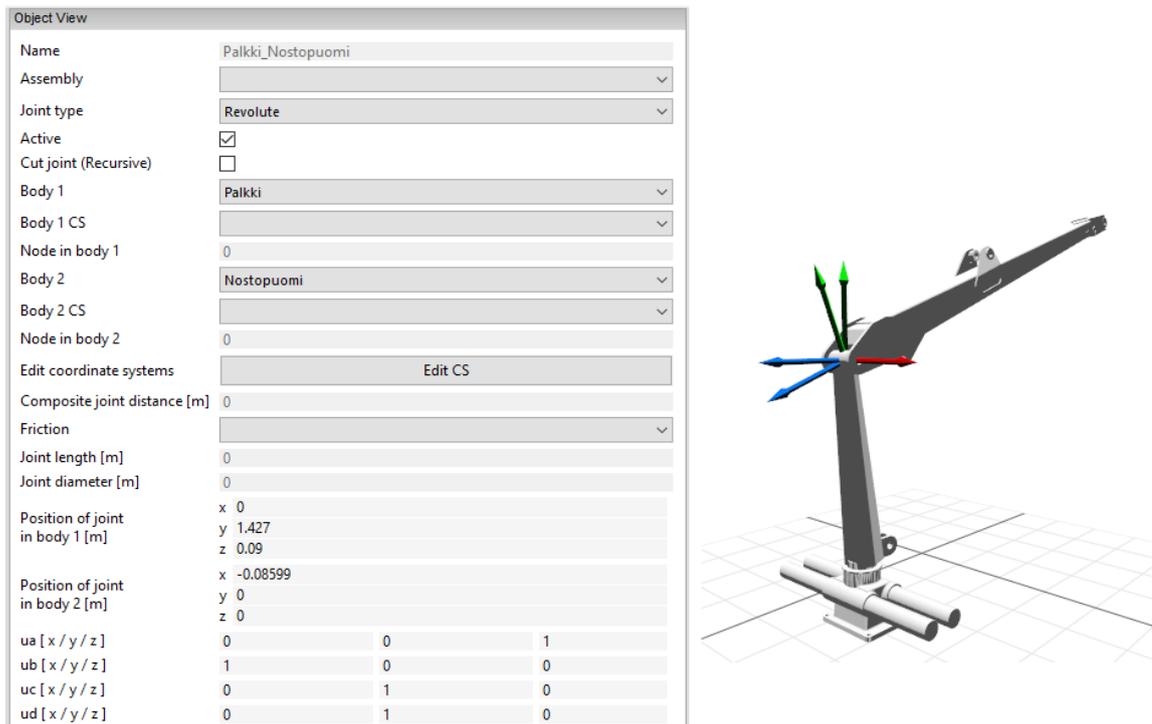


Figure 4. Revolute joint between pillar and lift boom.

If the constraints are defined correctly the model should be a pendulum at this point. Since the system is now swinging freely, the next step is to add forces. Mevea has multiple different options for forces but in this thesis only translational forces were used. Translational forces are also called body-to-body force, meaning that the force acts between two selected positions as seen from the figure 5. In this case the translational forces are used to describe the hydraulic cylinders.

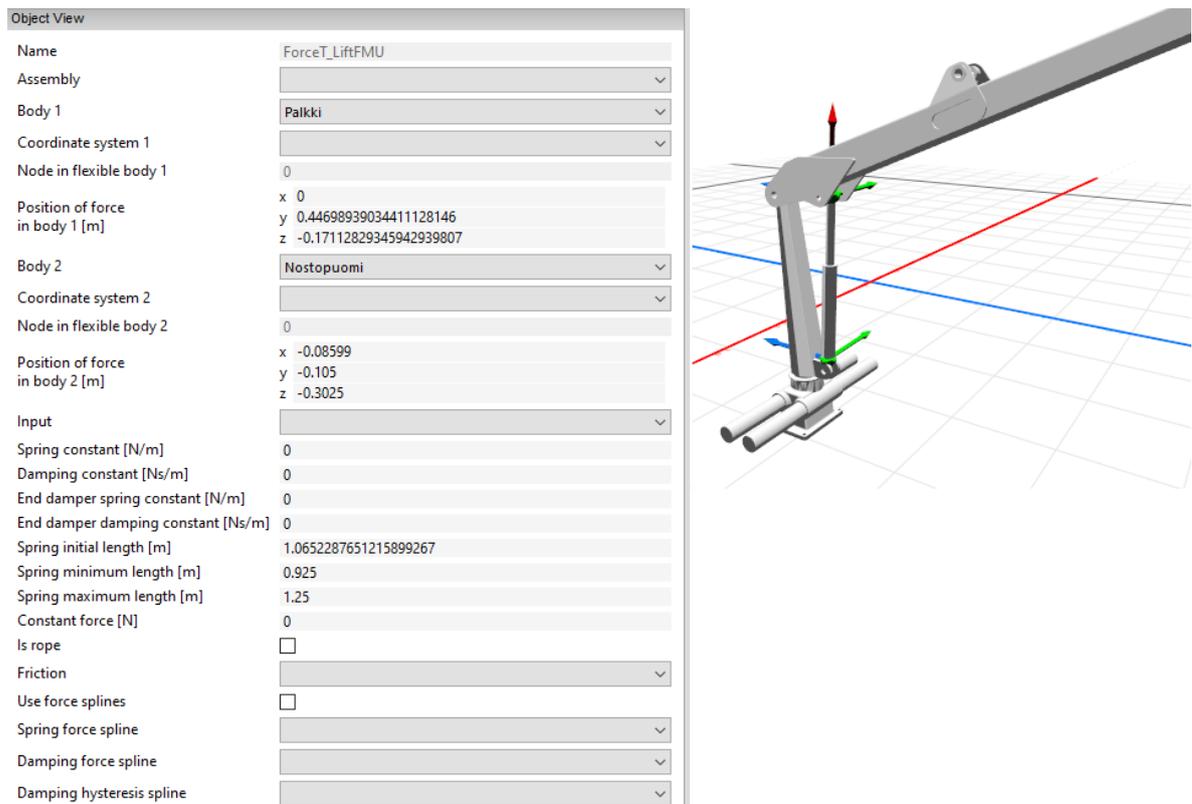


Figure 5. Translational force between pillar and lift boom.

The lift and tilt cylinders have been modeled as “dummies” in this study. According to Mevea “Dummies are parts which are not connected to the kinematic chain, but affect the mass and inertia properties of the attachment body” (Mevea Modeller: Beginner tutorials, p. 19). Basically they are not normal bodies, but their mass properties affect the bodies they are attached to. Hydraulic cylinders are modeled as dummies because this method reduced the number of bodies and constraints and thus makes the simulation lighter. Dummies are defined similarly to normal bodies but they also require selecting dummy type and dummy index. In this model cylinder forces were modeled as body-to-body forces which means that dummy types must also be body-to-body forces. (Reference Manual for Solver Library, p. 15.)

At this point the mechanics are finished (see figure 6) and normally the process would continue by creating controls and hydraulics in Mevea. In this case the hydraulics and control signal are made with Simulink and imported to Mevea with FMU, so the final step is to prepare the model for FMU. This includes adding outputs for cylinder positions and velocities, and inputs for cylinder forces as seen from figure 7.



Figure 6. Patu-655 model in Mevea Modeller.

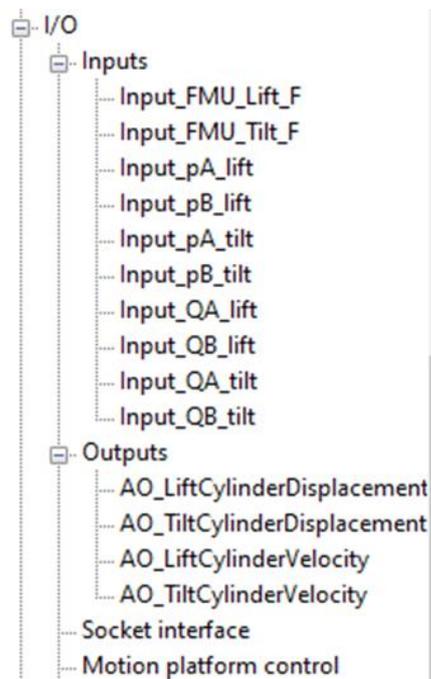


Figure 7. List of inputs and outputs in Mevea.

2.3 Hydraulics model in Simulink

Simulink is used for modeling the Patu-655 hydraulic circuit. The hydraulic circuit consists of two hydraulic cylinders, two directional control valves (DCV) and load sensing pump as seen from the figure 8. The hydraulics won't be explained in detail because the hydraulics in this model are exactly the same as in Shevchuk (2020) thesis and more information is available there.

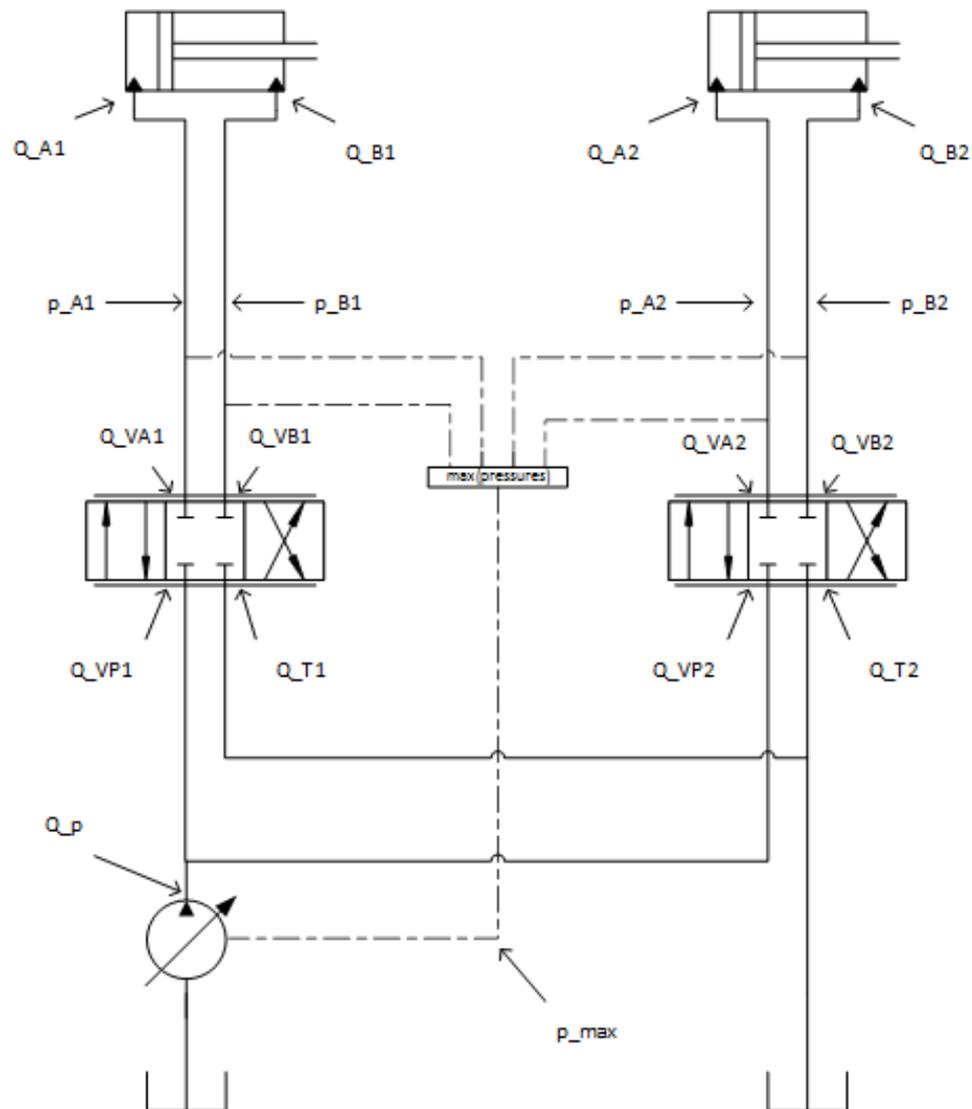


Figure 8. Hydraulic circuit.

2.3.1 Directional control valve equations

Position of valve spool follows input signal with a delay so the feedback signal U is calculated with following equation:

$$\dot{U} = \frac{U_{in} - U}{\tau_p} \quad (1)$$

, where U_{in} is the input signal and τ_p is time constant of the valve.

The DCV is modelled based on Danfoss PVG 32. Volume flow equations required few additional conditions because when the pressure difference was less than 7 bar, the pressure compensator works as orifice. The volume flows for the directional control valve are calculated by following system of equations:

$$\begin{aligned} & \text{If } U \leq U_{lt} \\ Q_{VA} &= \begin{cases} Cv(U_{lt} - U)\sqrt{Dp} & \text{if } p_P - p_A \geq Dp \\ Cv(U_{lt} - U)\text{sign}(p_P - p_A)\sqrt{|p_P - p_A|} & \text{if } p_P - p_A < Dp \end{cases} \\ Q_{VB} &= Cv_T(U_{lt} - U)\sqrt{|p_B - p_T|} \\ Q_{VP} &= Q_{VA} \end{aligned} \quad (2)$$

$$\begin{aligned} & \text{If } U \geq U_{ut} \\ Q_{VA} &= -Cv_T(U - U_{ut})\sqrt{|p_A - p_T|} \\ Q_{VB} &= \begin{cases} -Cv(U - U_{ut})\sqrt{Dp} & \text{if } p_P - p_B \geq Dp \\ -Cv(U - U_{ut})\text{sign}(p_P - p_B)\sqrt{|p_P - p_B|} & \text{if } p_P - p_B < Dp \end{cases} \\ Q_{VP} &= -Q_{VB} \end{aligned} \quad (3)$$

, where Q_{VA} and Q_{VB} are volume flows for valve lines A and B, Q_{VP} is volume flow for valve pump line, Cv is semi-empirical flow rate coefficient for flow from pump to line A or B and Cv_T is semi-empirical flow rate coefficient for flow direction from lines A or B to tank, U_{lt} and U_{ut} are valve opening thresholds voltages, $Dp = 7 \cdot 10^5 Pa$ is compensator pressure, p_P , p_A , p_B and p_T are pump, A port, B port and tank pressures respectively.

2.3.2 Load sensing pump equations

Load sensing pump volume flow Q_P is calculated with following equation:

$$\dot{Q}_P = \frac{K_P (\Delta p_{ref} - (p_P - \max(p_1, p_2 \dots p_n))) - Q_P}{\tau_p} \quad (4)$$

, where $K_P = 1 \cdot 10^{-9}$ is pump semi-empirical constant, $\Delta p_{ref} = 35 \cdot 10^5 Pa$ is pump controller reference pressure, $\max(p_1, p_2 \dots p_n)$ is the highest cylinder chamber pressure and $\tau_p = 0.03 s$ is pump time constant. The maximum pressures are used for controlling the pump, so that the pump pressure will be a bit higher than the highest pressure in the system. Pump pressure p_P is calculated with equation:

$$p_P = \frac{B_{epI}}{V_{pl}} (Q_P - Q_{VP}) \quad (5)$$

, where $B_{epI} = 1300 \cdot 10^6 Pa$ pump bulk modulus of the pump, $V_{pl} = 3 \cdot 10^{-3} m^3$ is volume of hose connected from pump to valve, Q_P pump volume flow and Q_{VP} volume flow from valve to cylinder. Equation 4 and 5 won't be used in the final version of the model because laboratory measurements for the pump pressure will be used instead.

2.3.3 Hydraulic cylinder equations

The force produced by the cylinder F is calculated by using chamber pressures and cylinder areas:

$$F = p_A A_A - p_B A_B - F_\mu \quad (6)$$

, where A_A is cylinder piston side area, A_B is cylinder rod side area and F_μ is cylinder friction force. The values needed for calculating cylinder areas and volumes are listed in table 1.

Table 1. Patu-655 cylinder dimensions.

	Lift cylinder	Tilt cylinder
Cylinder side diameter (mm)	100	90
Rod side diameter (mm)	56	56
Range (mm)	0-535	0-780

Cylinder friction force F_μ is calculated with following formula because it also takes cylinder velocity into account (Malysheva et al. 2018, p. 2 & Andersson et al. 2007, p. 583)

$$F_\mu = F_C \tanh\left(\frac{v}{v_C}\right) + bv \quad (7)$$

, where $F_C = 900N$ is Coloumb friction, v is cylinder velocity, $v_C = 0.0001m/s$ is Coloumb velocity threshold and $b = 4000 NS/m$ is viscose friction coefficient.

Cylinder volume flows are calculated with following equations:

$$Q_A = A_A v \quad (8)$$

$$Q_B = -A_B v \quad (9)$$

, where A_A is cylinder piston side area and A_B cylinder rod side area.

Cylinder volumes are calculated with:

$$V_A = A_A y + V_h \quad (10)$$

$$V_B = A_B(H - y) + V_h \quad (11)$$

, where y is cylinder position, V_h is hose volume from valve to cylinder, H is cylinder maximum stroke.

Cylinder pressures are calculated with equations 12 and 13:

$$\dot{p}_A = \frac{B_{eA}}{V_A + V_H} (Q_{VA} - Q_A) \quad (12)$$

$$\dot{p}_B = \frac{B_{eB}}{V_B + V_H} (Q_{VB} - Q_B) \quad (13)$$

, where B_{eA} and B_{eB} are effective bulk modules of cylinder chambers A and B.

In Shevchuk (2020) study the effective bulk modules were calculated with following equations because the amount of air in the hydraulic liquid was considered to have more impact than hydraulic hoses:

$$B_e = \left(\frac{1}{B_{oil}} + \frac{V_{air}}{(V_{air} + V_C)B_{air}} \right)^{-1} \quad (14)$$

, where $B_{oil} = 1.5 \cdot 10^9 Pa$ is bulk modulus of oil, $V_{air} = 2.5 \cdot 10^{-5}$ volume of air in the hydraulic fluid, V_C cylinder chamber volume, $B_{air} = 1.4p$ bulk modulus of air and p cylinder chamber pressure.

2.4 Functional Mock-up Unit

According to Modelica Association "FMI is a tool-independent standard to support both model exchange and co-simulation of dynamic models using a combination of XML files and C code (either compiled in DLL/shared libraries or in source code)" (Modelica Association 2020, p. 1). FMU is a ZIP file, which consists of FMI description file in XML format, source code of the model in C language with standardized functions and possible additional FMU data. The description file includes the data used in the FMU and defines the variables in standardized way. The C functions are used for the equations in the model. (Modelica Association 2020, p. 4.) Simplified, this means that FMU contains the simulation model, and the FMI connects the FMU to other simulation software.

The basic idea behind FMI is to allow using components by different suppliers simultaneously. This is great for product development and researching, because usually different software excels in different areas and if you stick with one software, you usually have to make some tradeoffs. However, with FMUs this problem is avoided, and you can use different tools for different areas. For example, in this thesis Mevea was used to model mechanics and Simulink for hydraulics and the Simulink model was exported as FMU and then imported to Mevea.

FMI can be divided into two categories: FMI for Model Exchange and FMI for Co-Simulation. The largest difference between these two is that Model Exchange doesn't include numerical integrator (solver) and the Co-Simulation includes its own solver, as you can see from figure 9 and 10. However, these two operate similarly because both of them use standardized C functions for the equations and both of them include the FMI description file. This information is enough for basic user, but more detailed information is available at FMI standard. (Modelica Association 2020, p. 9.)

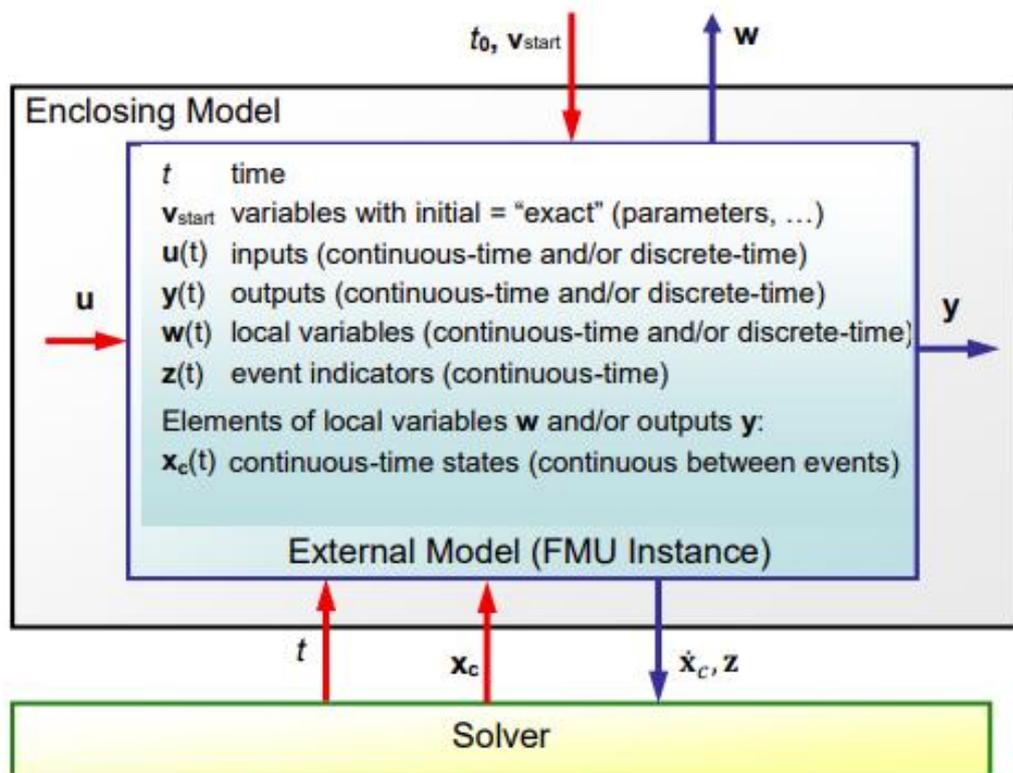


Figure 9. FMI for Model Exchange (Modelica Association 2020, p. 73).

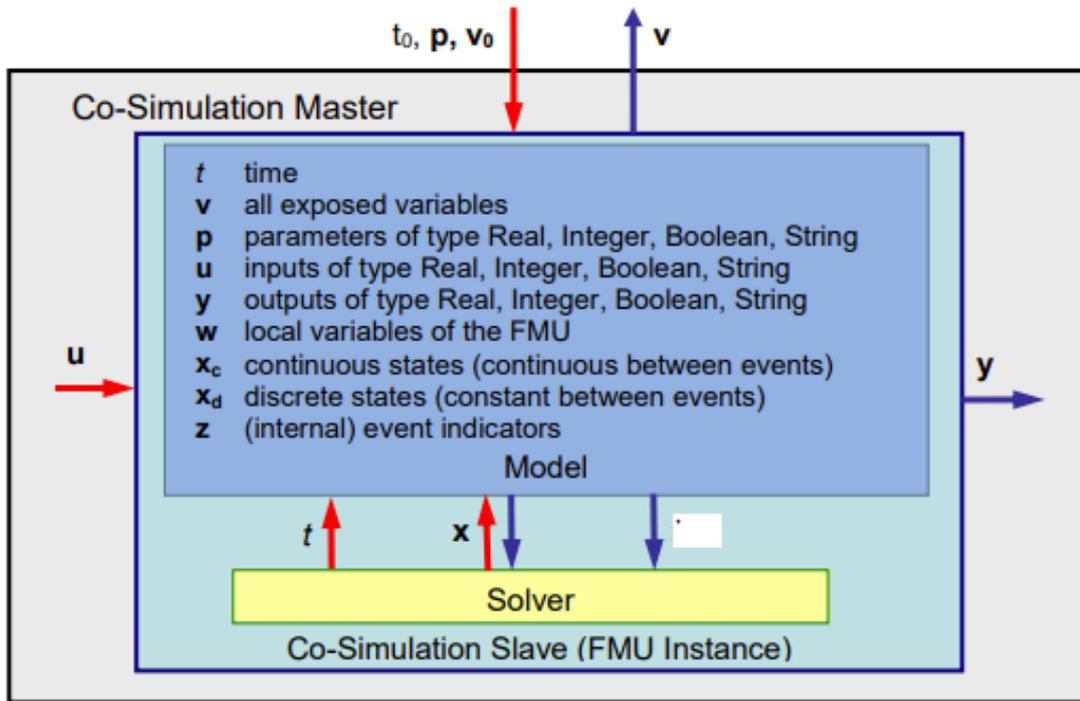


Figure 10. FMI for Co-Simulation (Modelica Association 2020, p. 98).

As you can see from figures 9 and 10, information is provided to the FMU and it also outputs information. For example, in figure 10, the Co-Simulation Master software inputs $u(t)$ to the Co-Simulation Slave (subsystem) and the FMU interacts with the input and calculates the equations with its solver and then outputs $y(t)$ back to the Co-Simulation Master. This interaction happens at least once in Co-Simulation Master's timestep as seen from figure 11 left side. The FMU model in this thesis uses the method on the right side of figure 11 because hydraulic equations often need smaller timestep than multibody solver due to numerically stiff equations. FMU in this system uses 0.25ms timestep for hydraulics and Mevea uses 1ms timestep for mechanics, because the hydraulics are relatively simple, and this produces sufficiently accurate results. However, if the hydraulic circuit was more complex and had more small volumes, the FMU timestep would be lower, for example 0.1ms. (Rahikainen et al. 2018, p. 294 & Kiani-Oshtorjani 2020, p. 24.)

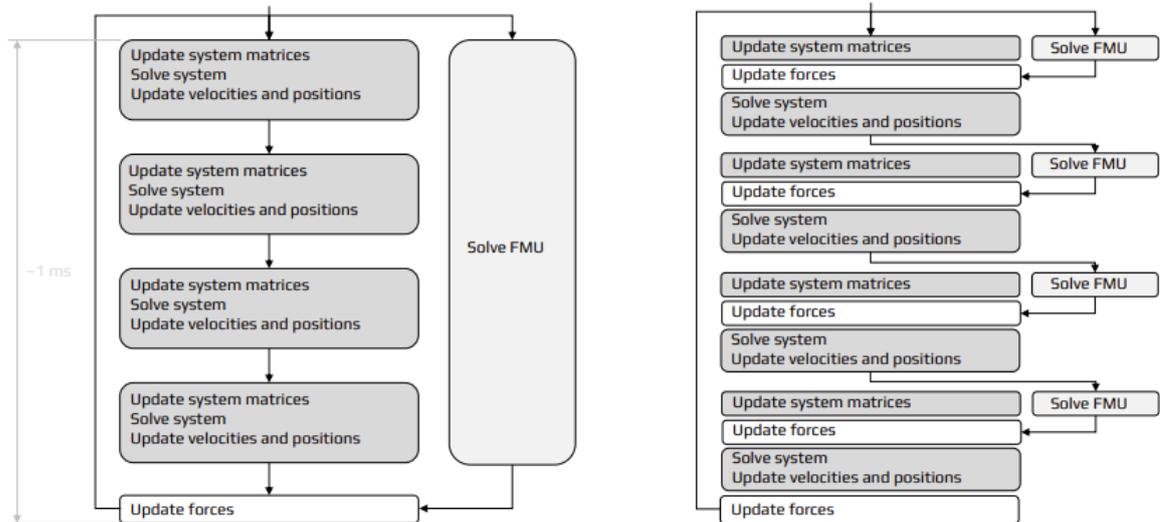


Figure 11. Different ways to solve FMU (Rouvinen, p. 7).

Mevea supports FMI for Co-Simulation based on FMI standard 2.0, which means that the Co-Simulation will be explained more in depth. The FMI for Co-Simulation can also be divided into two categories: Co-Simulation with generated code (see figure 12) and Co-Simulation with tool coupling (figure 13). The main difference between these two is that with tool coupling all of the simulation software must also be running, but with generated code only the master simulation program is running. Tool coupling also uses FMI Wrapper to connect FMI function calls to application programming interface (API) as seen from figure 13.

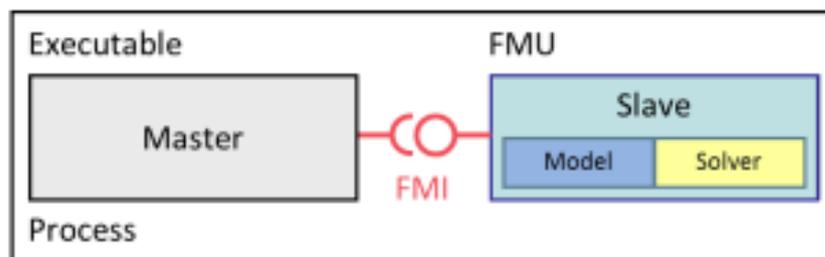


Figure 12. Co-Simulation with generated code (Modelica Association 2020, p. 97).

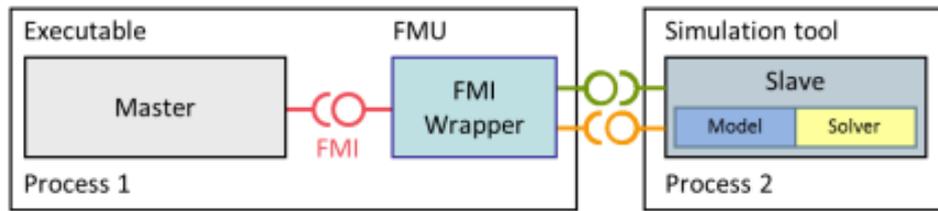


Figure 13. Co-Simulation with tool coupling (Modelica Association 2020, p. 97).

In this thesis, the generated code option was used because having to use multiple simulation software simultaneously leads to higher computation load without any benefits in this case. This also helps to achieve smaller timestep for the simulation with limited calculation power.

The final FMU has in total 6 inputs and 10 outputs, as seen from figure 14. The inputs are input voltage for lift and tilt cylinder, position, and velocity of the cylinder for both cylinders. The outputs are for cylinder forces, cylinder chamber pressures and cylinder volume flows. Only the force outputs are actually used in the Mevea, but rest of them were used for troubleshooting and monitoring how the FMU performs. Also, the control signal was imported from the laboratory measurements, but if needed it can be connected directly to Mevea for further studies.

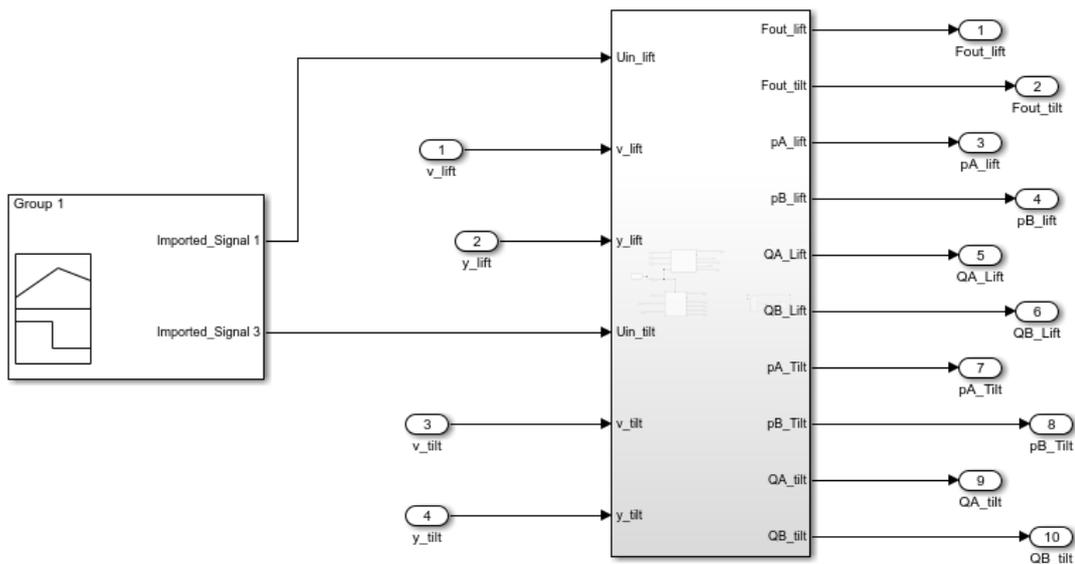


Figure 14. Inputs and outputs in Simulink model.

The FMU from figure 14 works just like figure 10 shows, meaning that Mevea is Co-Simulation Master and Simulink model is the slave which interacts with the cylinder position and velocity with equations from chapter 2.3 and finally outputs the cylinder forces. And as stated before, the Simulink model was exported as Co-Simulation Standalone FMU, which is the same thing as Co-Simulation with generated code and it uses Runge-Kutta 4 solver with 0.25 ms timestep. After that the FMU is imported to Mevea, and the inputs and outputs are connected to the corresponding ports. Before the system is usable, the user has to verify if both systems use same units for parameter values and if any signal conversions are needed. For this model only the initial cylinder displacement was calculated differently in Simulink than in Mevea, and this was fixed by adding offset to the cylinder position output from Mevea. For other models this final step could be for example conversion of motors angular velocity from rad/s to RPM, changing displacement unit from meters to millimeters or changing directions for forces or other values.

3 RESULTS

The simulation was divided into two parts, one for testing the lift cylinder and the other for tilt cylinder. Both of these simulations are compared to Shevchuk (2020) thesis simulation results and laboratory measurements. Parameter values and equations are also same as in Shevchuk (2020) study, so that the results are comparative. Both of the simulations used the same solver settings which can be seen from figures 15 and 16, but they had some different starting values which are introduced in following chapters. Both systems are simulated with 140 bar pump pressure and 201 kg mass at the end of the tilt boom.

Solution method	Recursive
Numeric solver	Runge-Kutta 4
Use sparse solver	<input checked="" type="checkbox"/>
Time step [s]	0.001
Simulation Time [s]	67.1
Hydraulics integrator	Numeric solver
Hydraulics integration steps	10
Oil bulk modulus	1.6e9
Enable hydraulics cavitation	<input type="checkbox"/>
Update frequency	0
Solve constraint forces with recursive method	<input type="checkbox"/>
Gravitation [m/s ²]	x 0
	y -9.81
	z 0
Integrator error tolerance	0.001
Maximum number of iterations	3

Figure 15. Mevea solver settings.

Simulation time

Start time: Stop time:

Solver selection

Type: Solver:

▼ Solver details

Fixed-step size (fundamental sample time):

Figure 16. FMU solver settings.

3.1 Lift cylinder simulation

The DCV for lift cylinder uses a spool with flow rate of 40 l/min with maximum opening. The spool is in lifting position when the control signal is 5.56V or lower and in lowering position when the signal is 6.44V or higher. The parameter values used for lift cylinder simulation are listed in table 2. The control signal for lift cylinder simulation is represented in figure 17.

Table 2. Parameter values for lift cylinder simulation.

Parameter	Value
Cv	$3.3201 \cdot 10^{-7} \frac{m^3}{sV\sqrt{Pa}}$
$pA_initial$	$63.471 \cdot 10^5 Pa$
$pB_initial$	$1 \cdot 10^5 Pa$
Cv_T	$4.0373 \cdot 10^7 \frac{m^3}{sV\sqrt{Pa}}$
Vh	$3.8003 \cdot 10^4 m^3$
p_max	$175 \cdot 10^5 Pa$

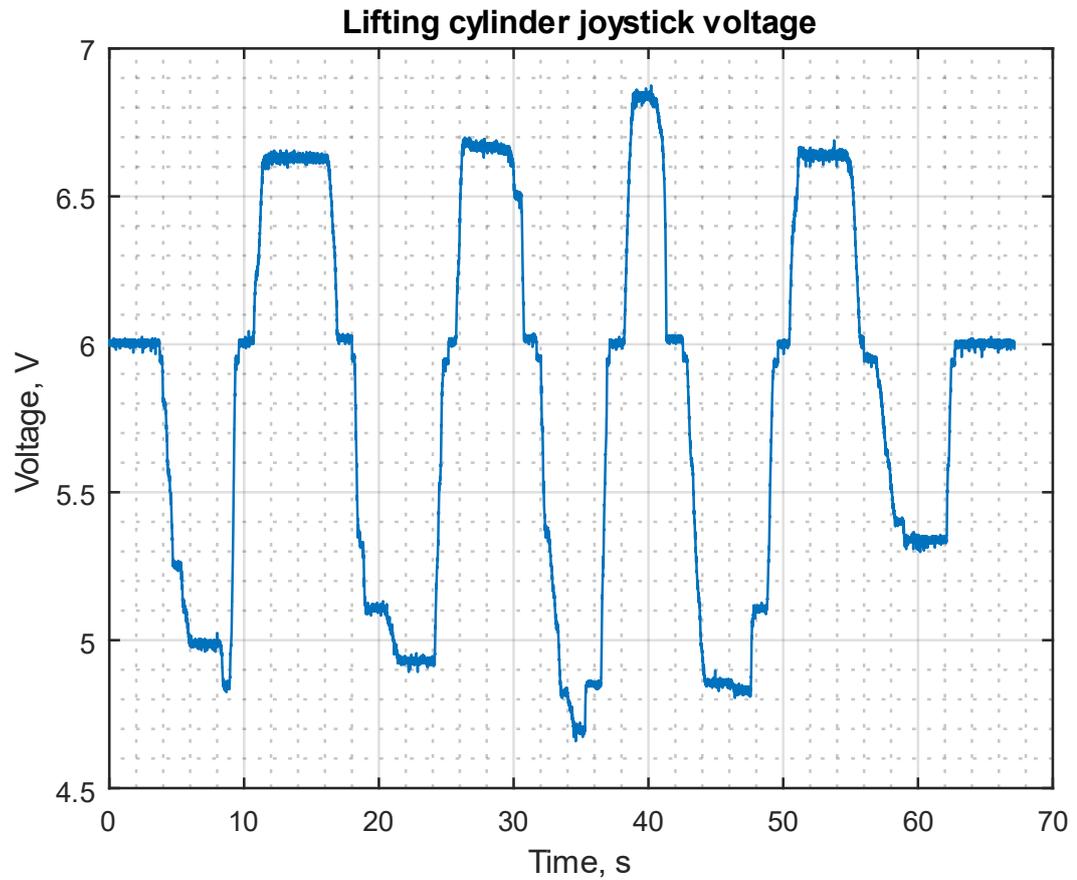


Figure 17. Input voltage for lift cylinder simulation.

As seen from figure 17, the lift cylinder is simulated for 67.1 seconds. The control signal from figure 17 results in up and down movement in lift cylinder as seen from 18. The control signal was imported from Shevchuk (2020) laboratory measurements.

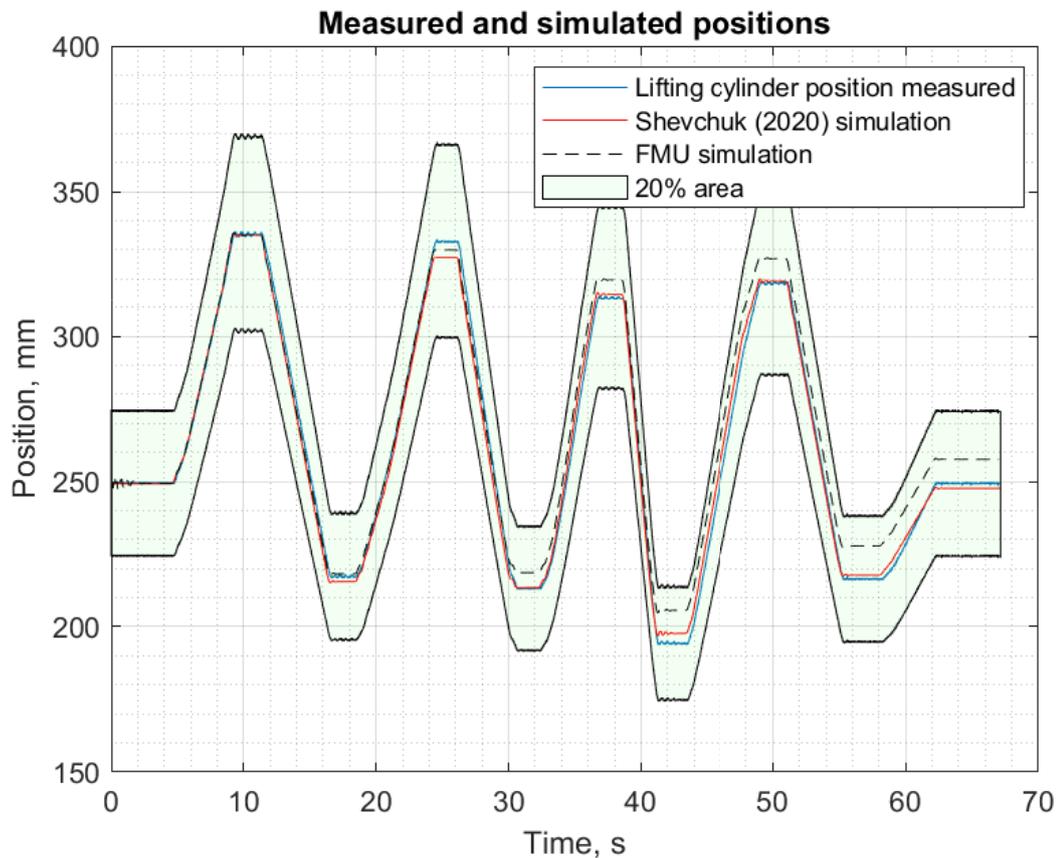


Figure 18. Simulated and measured positions.

In figure 18 the blue curve represents lifting cylinder position from laboratory measurements, the red one for Shevchuk (2020) simulation and the black one for FMU simulation. The FMU movement pattern is similar to the other two, but it starts to deviate at 30 seconds to the simulation, resulting in over 10 mm difference at 50 seconds. However, the FMU simulation does stay inside of the green area which represents 10% error margin to both directions. Maximum deviation compared to the measured position for the FMU model was 16.8 mm at 56 seconds and in Shevchuk (2020) simulation maximum deviation was 8 mm at 48 seconds.

Figure 19 represents simulated and measured cylinder chamber pressures. In the figure 19 red curve is lifting cylinder piston side pressure from laboratory measurements, blue is rod side pressure measured, yellow for piston side pressure from Shevchuk (2020), purple for rod side from Shevchuk (2020), black for piston side FMU simulation and magenta for rod side pressure from FMU simulation.

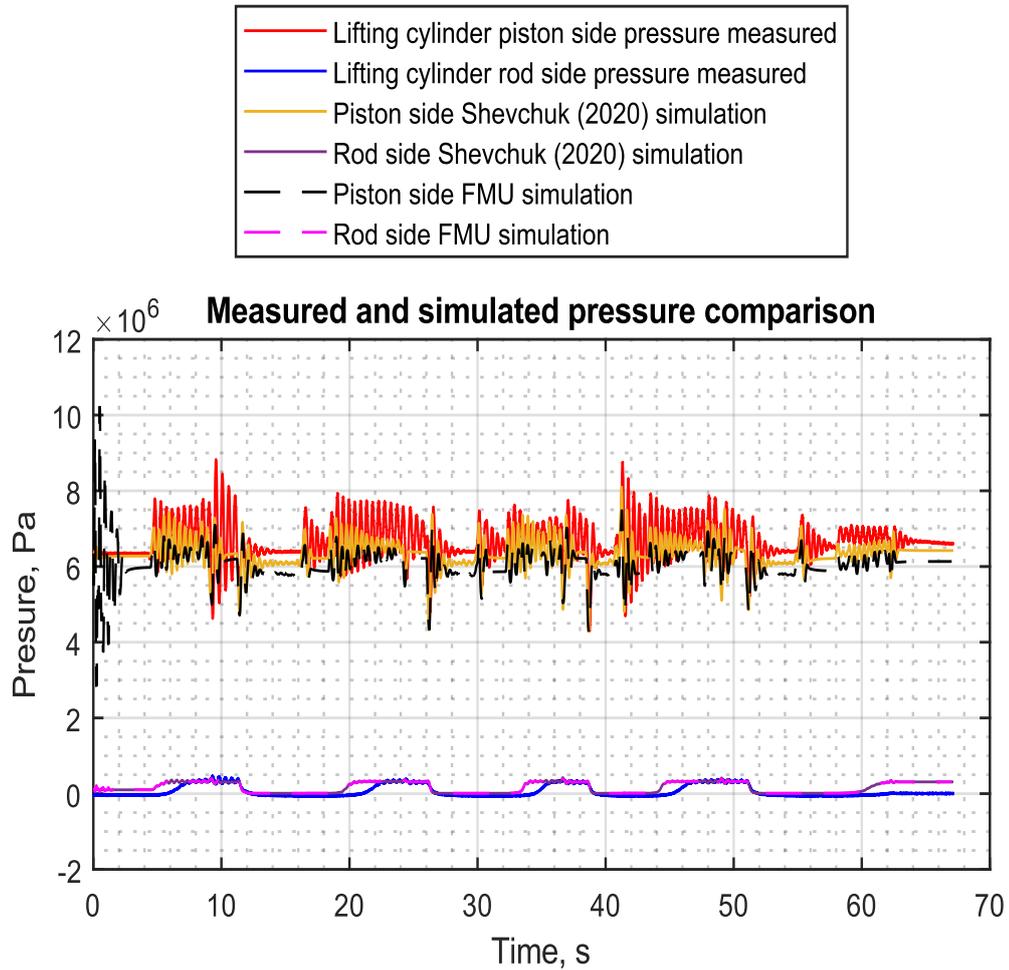


Figure 19. Simulated and measured lifting cylinder chamber pressures.

The FMU rod side pressure in figure 19 follows Shevchuk (2020) simulations almost perfectly and they both react to cylinder movements before the measured pressures, but all three of them have the same values. However, the piston side pressure with FMU has larger differences than rod side pressures. The FMU piston side pressure is constantly 2-5 bar lower compared to the measured pressure, but it does oscillate less than the measured pressure. Ignoring the pressure difference and oscillation at beginning, the FMU does act similarly to the measured and simulated pressures.

Figure 20 represents the measured and simulated lifting cylinder forces, where blue plot is for measured force, red for Shevchuk (2020) simulation force and yellow for FMU simulation force.

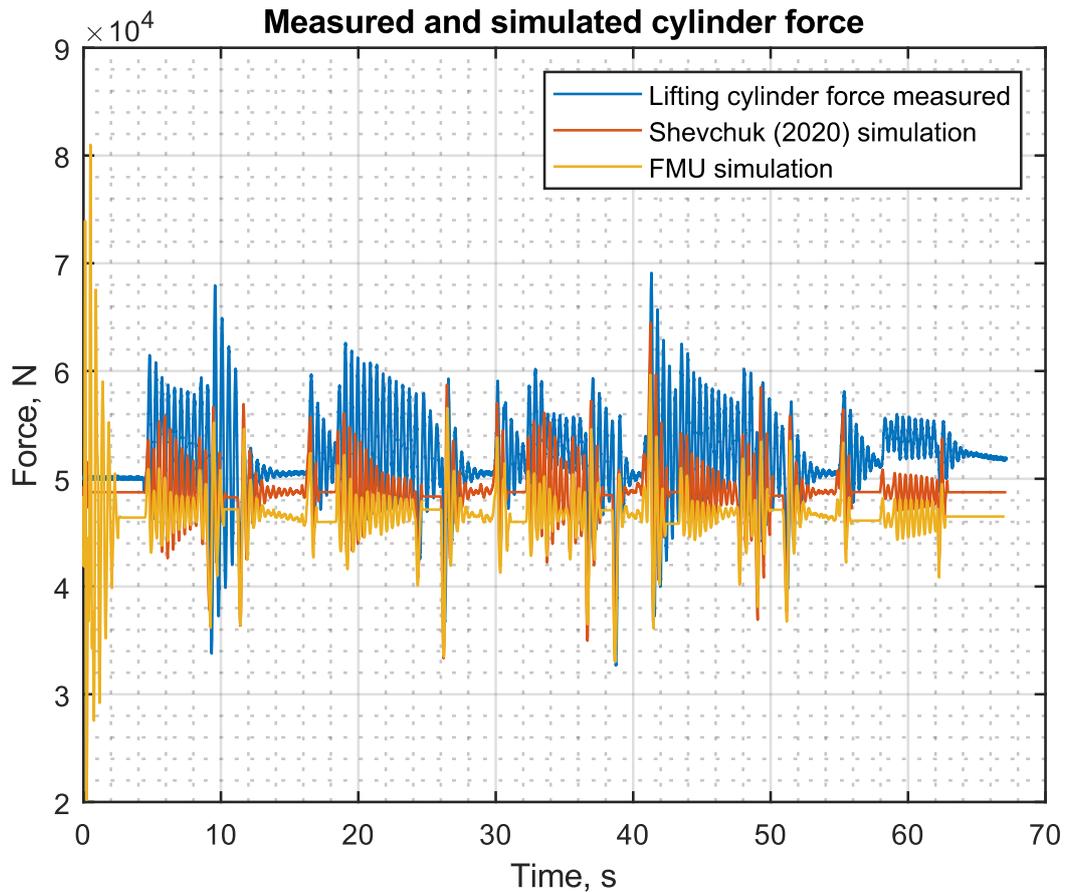


Figure 20. Simulated and measured lifting cylinder forces.

The FMU forces in figure 20 act similarly to simulated piston side pressures, it acts similarly to the measured force, but is constantly lower. Ignoring the oscillation at the beginning and the constant 4000 N difference, the FMU model can be considered as working correctly because the differences are small enough and the differences could be fixed by modifying parameter values or the mechanical model.

3.2 Tilt cylinder simulation

The DCV for tilt cylinder uses a spool with flow rate of 15 l/min with maximum opening. The spool is in lifting position when the control signal is 5.57V or lower and in lowering position when the signal is 6.22V or higher. Parameter values for tilt cylinder simulation are listed in table 3. The control signal used in tilt cylinder simulation is represented in figure 21.

Table 3. Parameter values for tilt cylinder simulation.

Parameter	Value
Cv	$1.245 \cdot 10^{-7} \frac{m^3}{sV\sqrt{Pa}}$
$pA_initial$	$50.14 \cdot 10^5 Pa$
$pB_initial$	$85 \cdot 10^5 Pa$
Cv_T	$1.0364 \cdot 10^{-7} \frac{m^3}{sV\sqrt{Pa}}$
Vh	$2.5335 \cdot 10^4 m^3$
p_max	$175 \cdot 10^5 Pa$

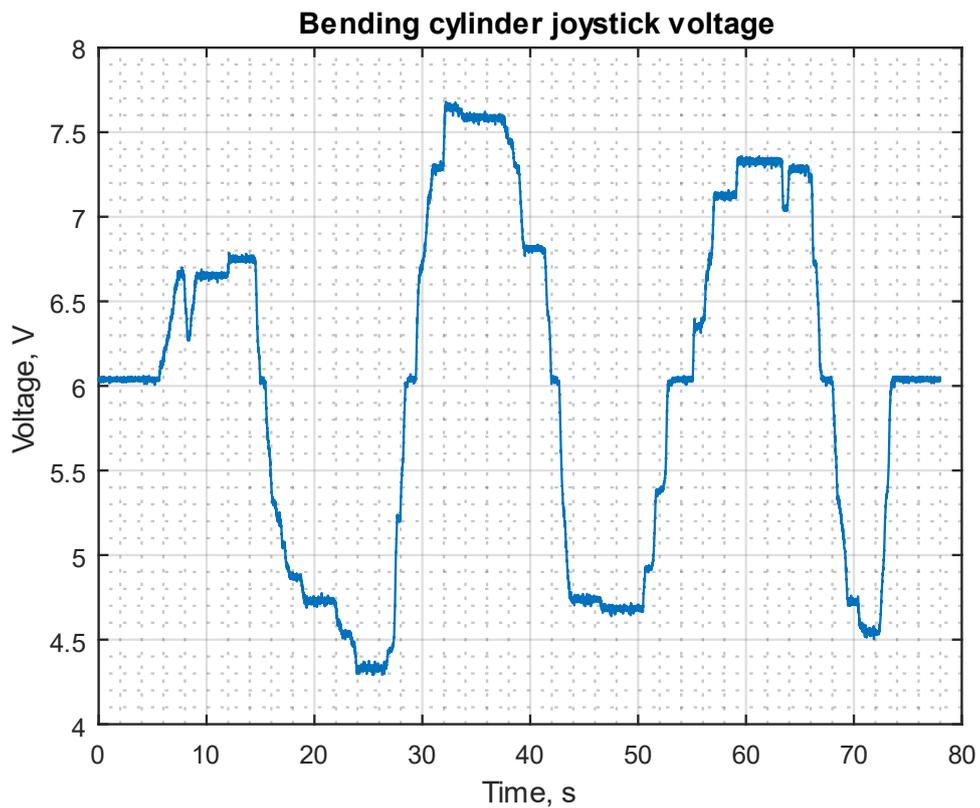


Figure 21. Input voltage for tilt cylinder simulation.

As seen from figure 21, the tilt cylinder simulation lasts 78 seconds and the cylinder is again simulated with simple up and down movement. However, the movements are much slower than with lift cylinder, as seen from figure 22.

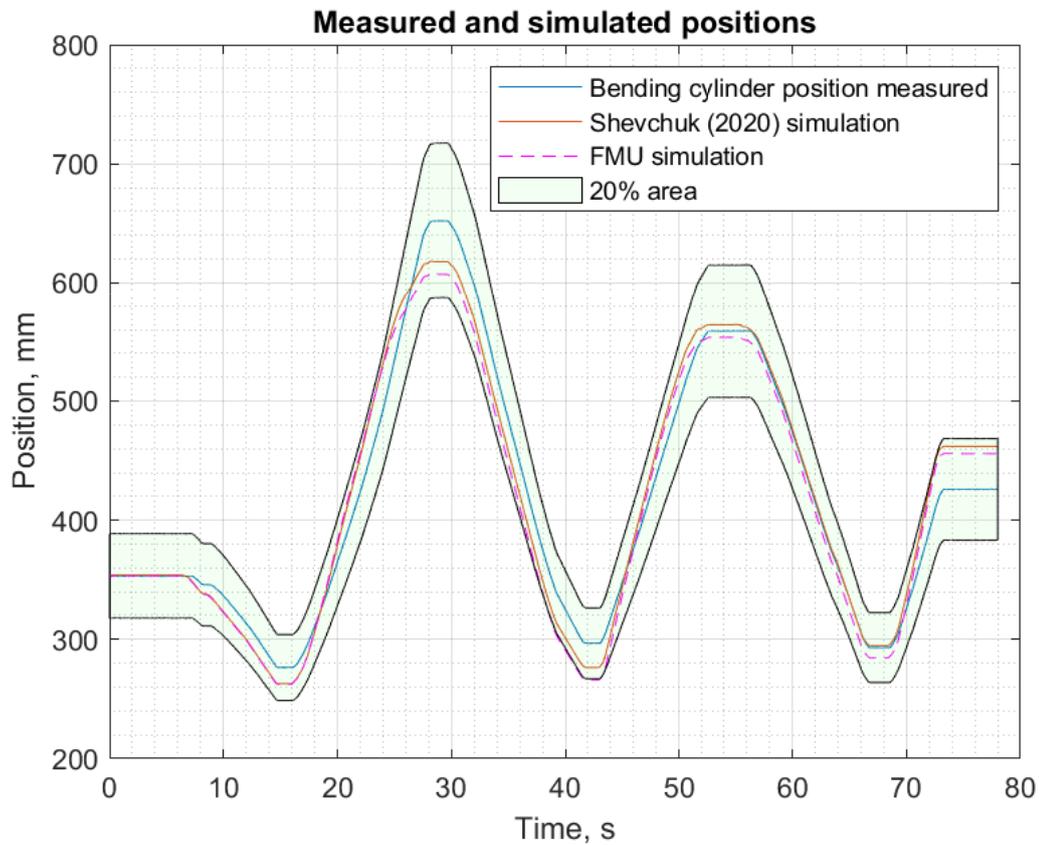


Figure 22. Simulated and measured tilt cylinder positions.

Figure 22 shows the simulated and measured cylinder positions, where the blue plot is the measured position, red Shevchuk (2020) simulation position and magenta FMU simulation position. As seen from figure 22, the FMU position follows the Shevchuk (2020) simulation position within 10 mm during the whole simulation, but both of them differ from the measured position during 20–45 seconds. During this time the largest difference is at the top position at 30 seconds when the difference is over 40 millimeters. Before 20 seconds and after 45 seconds the simulated models follow the measured position with small differences. Maximum deviation compared to the measured position for the FMU model was 45 mm at 29 seconds and in Shevchuk (2020) simulation maximum deviation was 36 mm at 78 seconds.

Figure 23 shows the simulated and measured tilt cylinder chamber pressures, where red plot describes tilt cylinder piston side measured pressure, blue rod side measured pressure, black Shevchuk (2020) simulation piston side pressure, cyan Shevchuk (2020) simulation rod side

pressure, green FMU simulation piston side pressure and magenta FMU simulation rod side pressure.

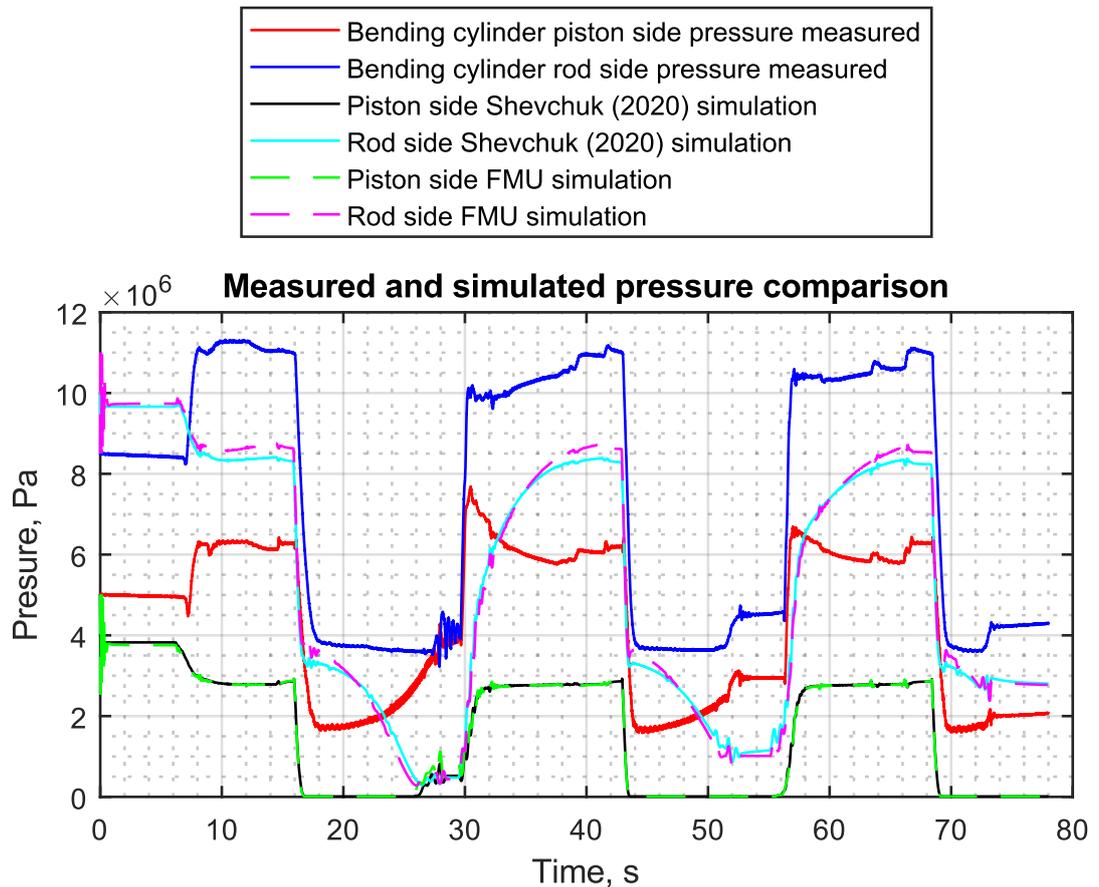


Figure 23. Simulated and measured pressures.

As seen from figure 23, the simulated pressures are almost similar, and the largest difference is less than 1 bar. However, both of the simulations have quite large differences compared to the measured ones. For example, at 35 seconds the simulated piston side pressures and measured pressure have over 30 bar difference, but they do act similar to the measured pressures. As seen from figure 24, the simulated and measured tilt cylinder forces don't have large difference, even though the pressures had.

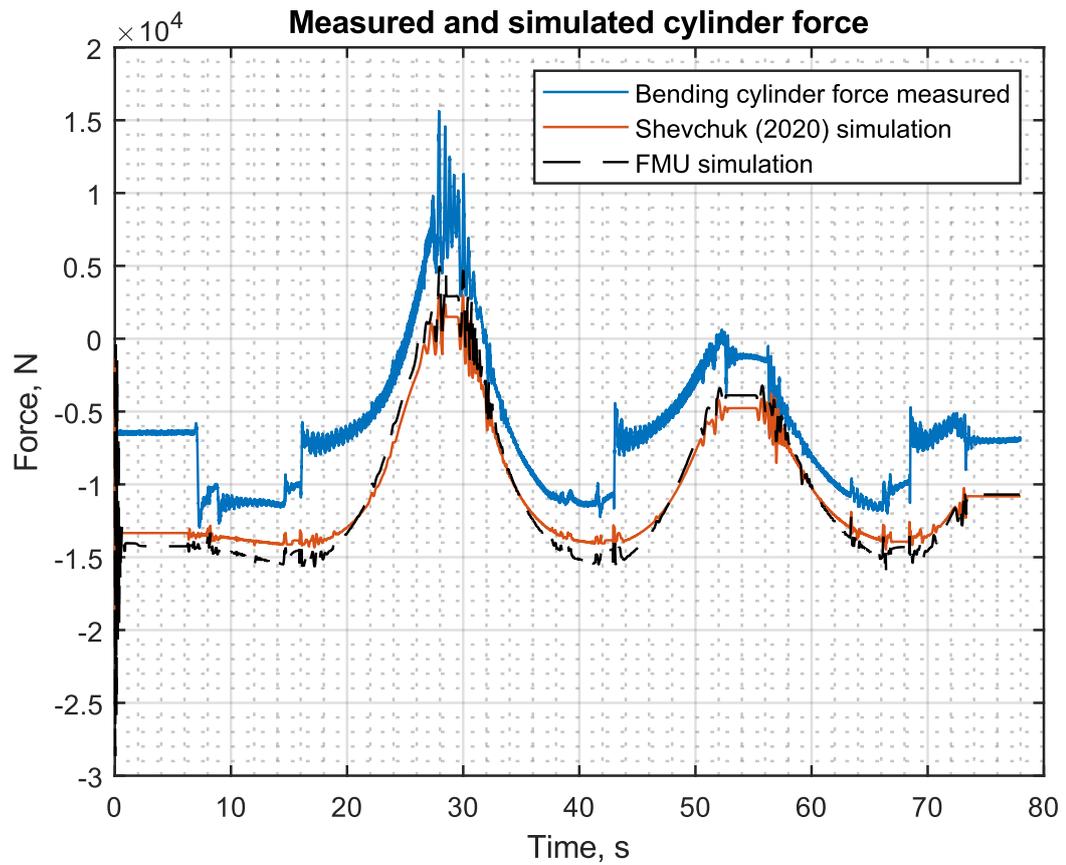


Figure 24. Simulated and measured tilt cylinder forces.

In the figure 24, blue represents measured tilt cylinder forces, red Shevchuk (2020) simulation force and black FMU simulation force. Most of the time, the simulated systems and measured one has close to 2000N difference, and all of them act similarly, just like with pressures and position. The FMU force has oscillation at the beginning of the simulation, but during the test, the measured one oscillates more than the simulated models. Even though, the forces and pressures are lower than the measured ones, the differences are small enough and the model can be considered working correctly.

4 DISCUSSION

This section compares the simulation results to previous research and discusses about the differences. First the cylinder position simulation results from this thesis is compared to laboratory measurements from Shevchuk (2020) and then the model developed in this thesis is compared to Simscape Multibody model from Shevchuk (2020) in terms of simulation speed. In addition to that, the simulation model is also built with Mevea's own tools for hydraulics and compared to the FMU model so that the benefits of building custom hydraulic circuit can be observed. The section ends with key findings and possible topics for further research.

4.1 Comparison to previous research

As seen from the results chapter, the FMU simulation model follows the Shevchuk (2020) simulation and measured data satisfactorily, as it behaves similarly but achieves lower values with pressures and forces, but the positions stay inside the 20% area with both simulations. These differences can be caused by multiple different reasons, for example differences in constrains, simplifications in the 3D-models, errors in hydraulic equations or parameters. Also, the differences between the FMU simulation and Shevchuk (2020) can be explained by the fact that they are made with different software. However, the differences are small enough for this thesis and the Mevea model can be developed further if needed. The cylinder position difference between FMU simulations and laboratory measurements from Shevchuk (2020) can be seen from figures 25 and 26.

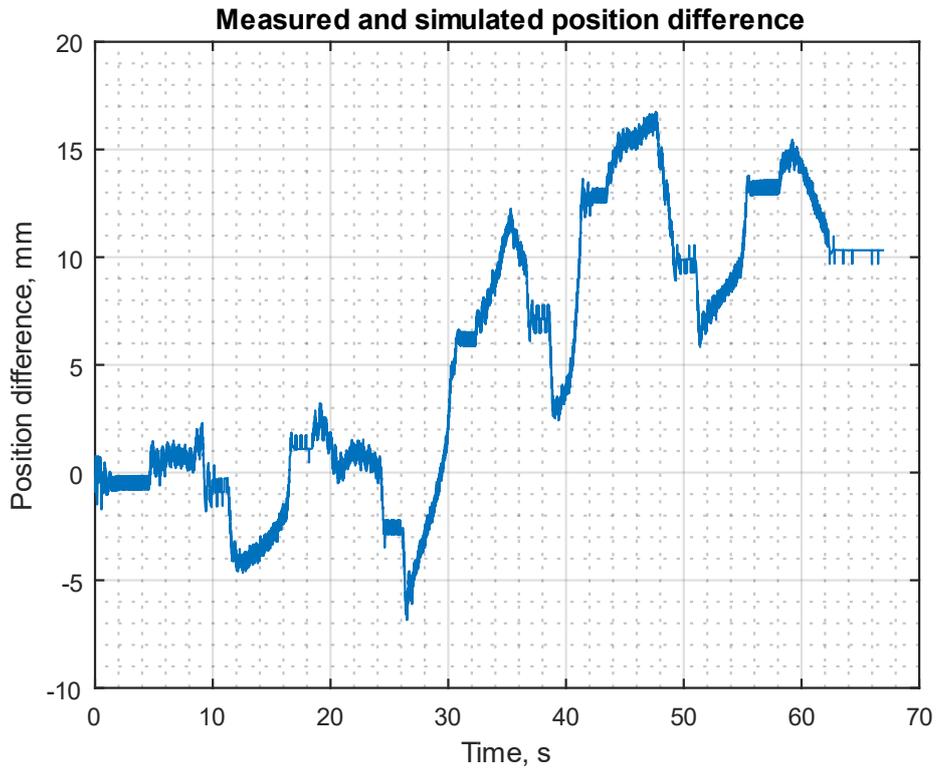


Figure 25. Position difference for lift cylinder simulation.

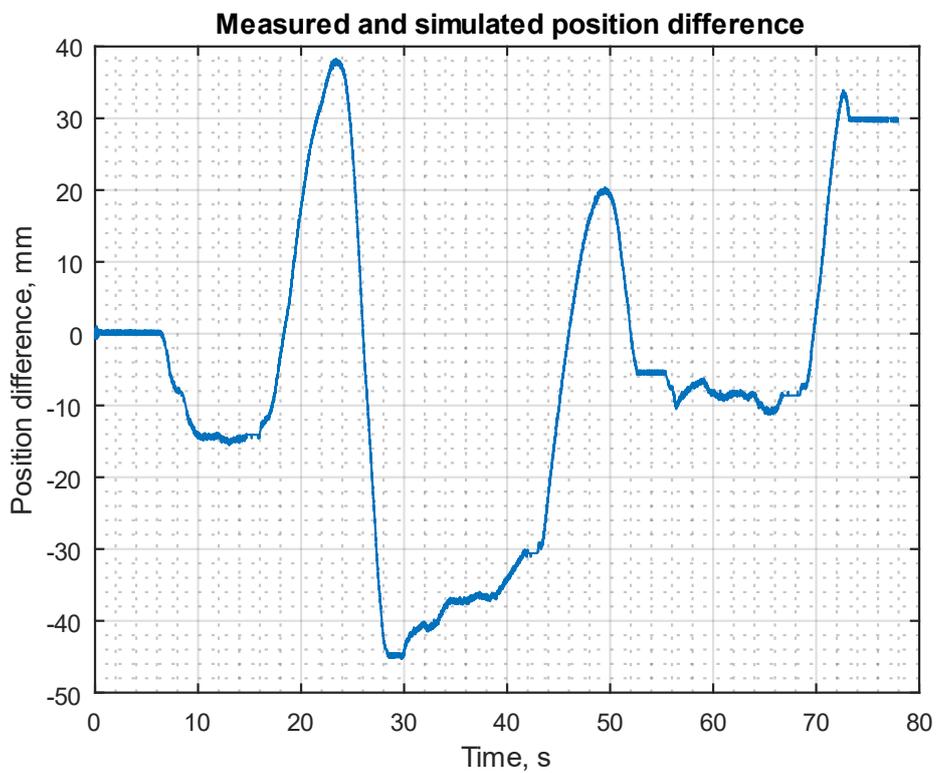


Figure 26. Position difference for tilt cylinder simulation.

As seen from figure 25 and 26, the deviation between measured and simulated lift cylinder position is below 15 mm during almost the whole simulation and it can be verified to be working correctly. However, as seen from figure 26, the deviation in tilt cylinder simulation is much higher. Although the tilt cylinder simulation does stay inside the 20% error margin area and can be considered as working, the simulation model could be revisited to make it more accurate. Most likely the issue is in the mechanical model at the 4-bar mechanism which connects the lift and tilt booms because rest of the tilt boom is modeled similarly to the lift boom and overall the model is very sensitive to the 4-bar mechanism accuracy. The issue could be caused for example by difference in joint types or joint positions in the 4-bar mechanism. Similar results were also found when the system was simulated with different pump pressures and control signals (see appendix I). These differences can also be caused by differences in parameter values in hydraulic circuit. For example adjusting valve opening thresholds or bulk modulus equations could make the system follow measured data better.

The largest difference between Shevchuk (2020) simulation model and the model developed in this thesis was the simulation time. The model built with Simscape Multibody was simulated with same solver settings as the model in this thesis, with timestep of 1 ms and Runge-Kutta 4, and with those settings the models run time for 100 second simulation was 15 minutes (Shevchuk 2020, p.51). In comparison, the model built with FMU was capable of real-time simulation as seen from figures 27 and 28.

The loop duration in figures 27 and 28 is the time required for simulating one timestep. For the simulation to run in real-time, the loop duration must be less than timestep. As seen from the figures, both systems are ready with the computations in less than 0.3 ms, excluding one small spike in tilt cylinder simulation, meaning that the simulation is running in real-time. However, it must be noted that the longest simulation with this model lasted 78 seconds, so it doesn't guarantee that the model will be able to achieve such low loop duration with longer simulations.

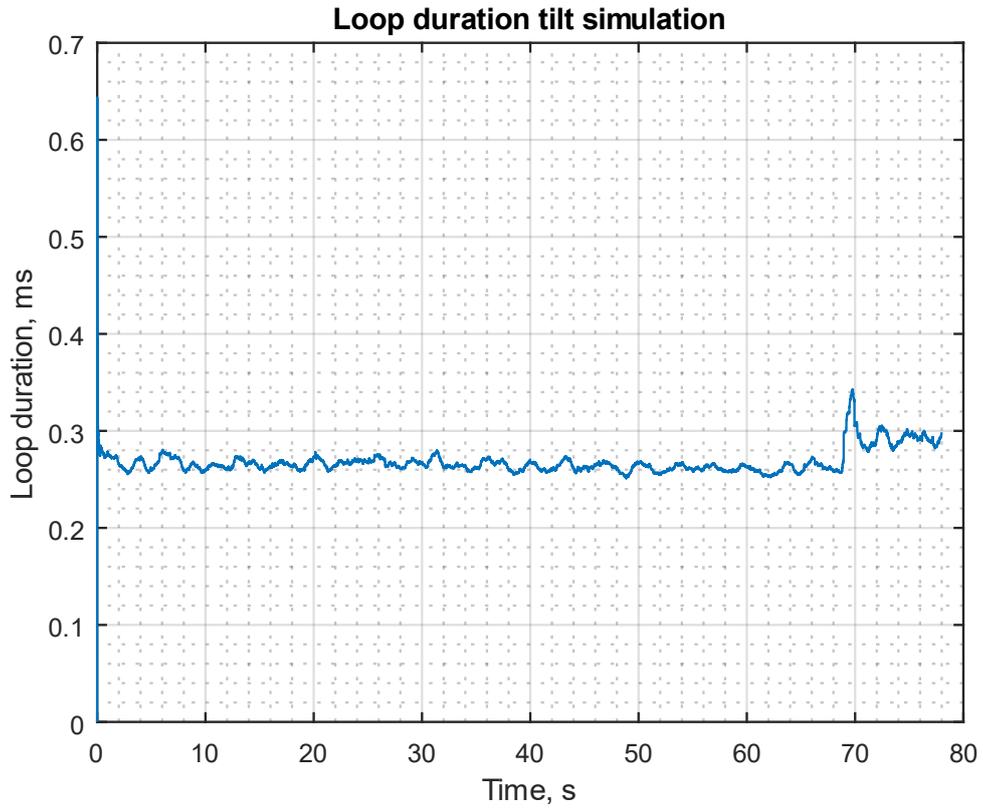


Figure 27. Loop duration for tilt cylinder simulation.

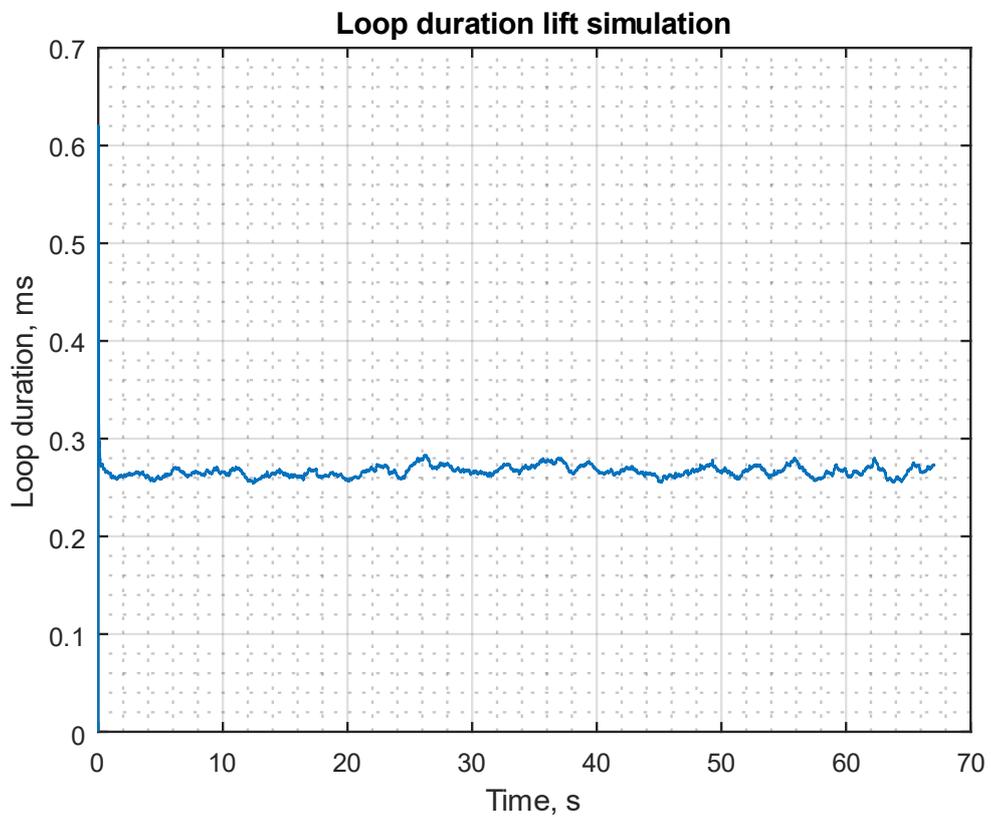


Figure 28. Loop duration for lift cylinder simulation.

In comparison to the Simscape Multibody model, the FMU model is significantly faster without any drawbacks. However, Shevchuk (2020) doesn't state the any specifications about the computer that was used for the simulation, and the FMU model was simulated on a computer with Intel Xeon Gold 6132 processor, so one reason for the drastic difference in simulation time could be the fact that one was simulated with low-end processor and another with higher-end processor. Also, the differences between software affect the simulation time because one of the Mevea's key feature is being able to simulate systems in real time, while that isn't necessarily the first priority with Simscape Multibody models.

Similar results can also be found from Malysheva (2021) and Mäkitalo (2020). Malysheva (2021) built a similar Patu-655 model with Simscape Multibody and used as a reference model. In that study the model was simulated for 5 seconds with Runge-Kutta 4 and time step of 0.1 ms. The 5 second simulation had a 7.49 second simulation time with Intel Core 2 Duo CPU @2.26 GHz and 4 GB of RAM. (Malysheva 2021, p. 47.)

Mäkitalo (2020) simulated Ponsse K121 loader, which is similar to a Patu-655, with combination of Mevea (mechanics), Amesim (hydraulics) and Simulink (control system). The model was built using the FMI for Co-Simulation principle, where Mevea acted as the master program and Amesim and Simulink as slaves. The connection between Mevea and Amesim was executed with generated code and with tool coupling between Mevea and Simulink. The Simulink model had 10 ms timestep, Amesim 0.25 ms, and Mevea 1 ms and used Runge-Kutta 4 solver. The system was simulated with 100 second and was simulated in real-time, while the largest loop duration was 0.93 ms with 1 ms timestep, with Intel i7-8850H processor. (Mäkitalo 2020, p. 35.)

In addition to software and computational power, solution method for expressing the model's dynamics affect results and simulation speed. Mevea offers two different formulations for multibody systems: recursive method and Lagrange, Penalty function. The difference between these two is that recursive formulation is computationally more efficient than Lagrangian formulation, as seen from figure 29.

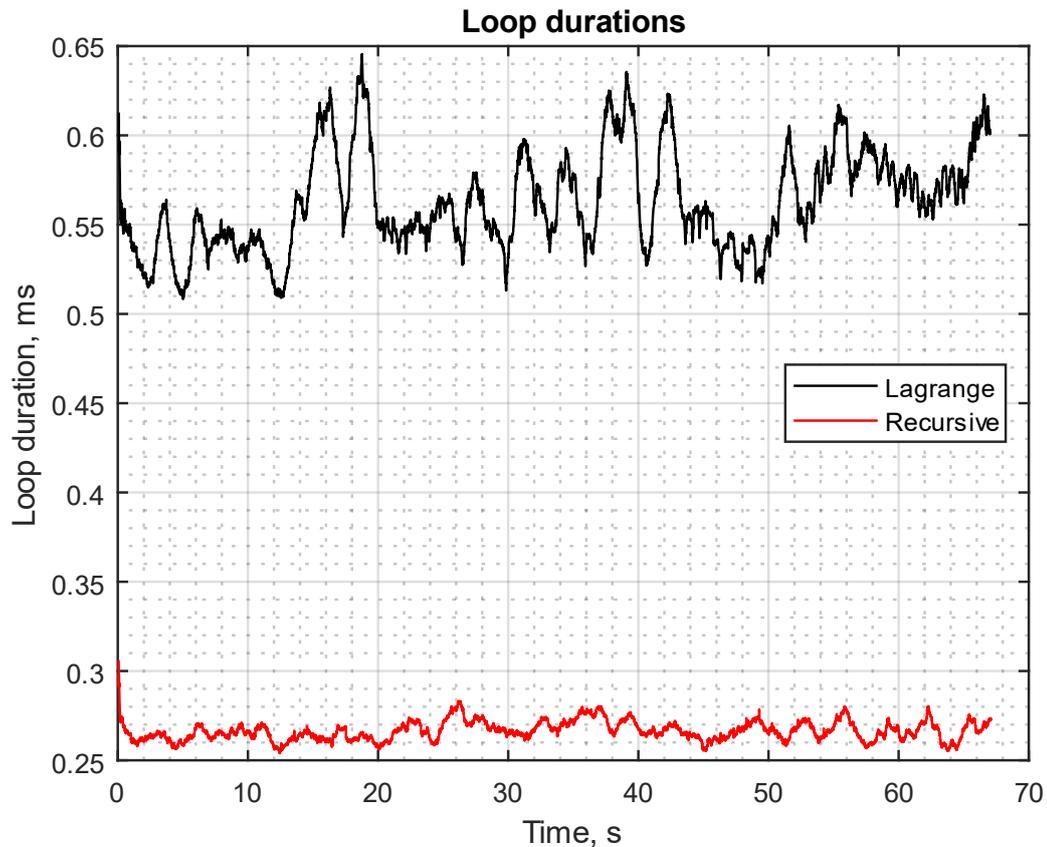


Figure 29. Loop durations with different multibody formalism.

However, the recursive formulation requires more precise joint definitions and Lagrangian formulation is more general option. For example with recursive method the joint order must be based on kinematic chain of the model and only certain joint types are available. In addition to that, closed kinematic loops require a cut joint to open one of the loops. In this model this means that one of the joints in the 4-bar mechanism must be a cut joint. (Reference Manual for Solver Library, p.11.) Both methods were able to run the simulation in real time but as seen from figure 29, the Lagrangian formulation was significantly slower than recursive formulation because, on average, the Lagrangian formulation required twice the time for simulating one timestep as the recursive formulation required.

4.2 FMU hydraulics compared to Mevea's hydraulics

The hydraulic circuit for Patu-655 was also built with Mevea, so that the difference between custom built hydraulic circuit and Mevea's hydraulics could be compared. Hydraulic circuit for this model was achieved by adding directional control valve, cylinder and volumes for pressure source, hoses and tank, and then connecting the corresponding components. These

ready-made components have enough parameters and customization for normal use, but the user for example can't add custom equations of parameters. For example, Mevea calculates effective bulk modulus of volume with equation 15 while in the custom circuit it was calculated based on the amount of air in the hydraulic circuit with equation 14.

$$B_{ei} = \left(\frac{1}{B_{oil}} + \sum_{j=1}^{n_c} \frac{V_j}{V_i + B_j} + \sum_{k=1}^{n_h} \frac{V_k}{V_i + B_k} \right)^{-1} \quad (15)$$

In equation 15 n_c is total number of hydraulic components related to volume i , V_j volume of single component, V_i combined volume of pipes, hoses and other hydraulic components, B_j bulk modulus of component j , n_h total number of pipes and hoses related to volume i , V_k constant volume of hose or pipe, B_k bulk modulus of hose or pipe. However, the oil bulk modulus takes the amount of air in oil into consideration and in this case the amount of air is approximately 0.5 %. (Reference Manual for Solver Library, p. 79.)

As seen from figure 30, the position difference for lift cylinder between custom made hydraulic circuit and Mevea hydraulics is insignificant while the cylinder is moving and when the cylinder isn't moving the model with Mevea's hydraulics is 10 mm higher on average. For this comparison the difference seems small but when comparing to the measured data, the position for Mevea hydraulics doesn't stay inside the 10 % error margin area.

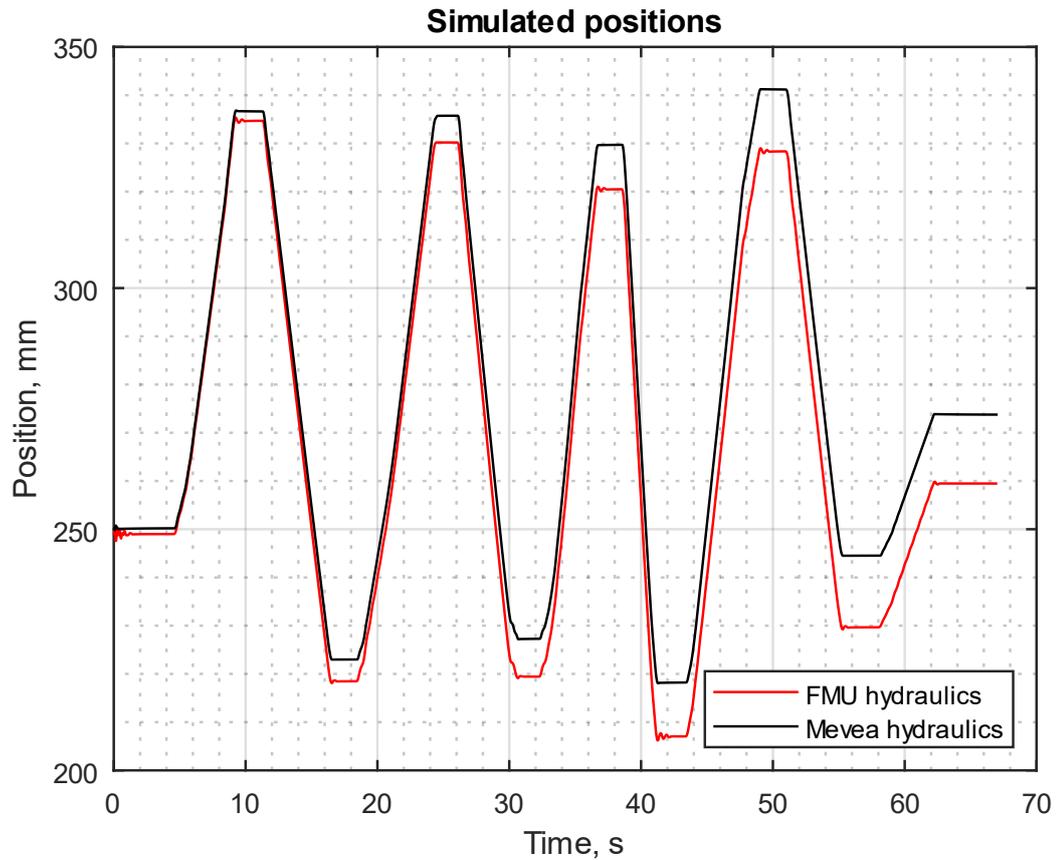


Figure 30. Simulated lift cylinder positions for FMU (red) and Mevea hydraulics (black).

As seen from figure 31, the Mevea hydraulics model follows the measured data satisfactorily until 30 seconds and after that it constantly deviates from the measured data about 10 % when the cylinder is moving. Based on the position results, the Mevea hydraulics model can't be verified to be working correctly and would need further developing so it would be more accurate. Although the model with Mevea's hydraulic circuit doesn't stay inside the 20 % area in terms of location, but the simulated pressures and forces act similar to the FMU model and achieve similar values (figures 32 and 33).

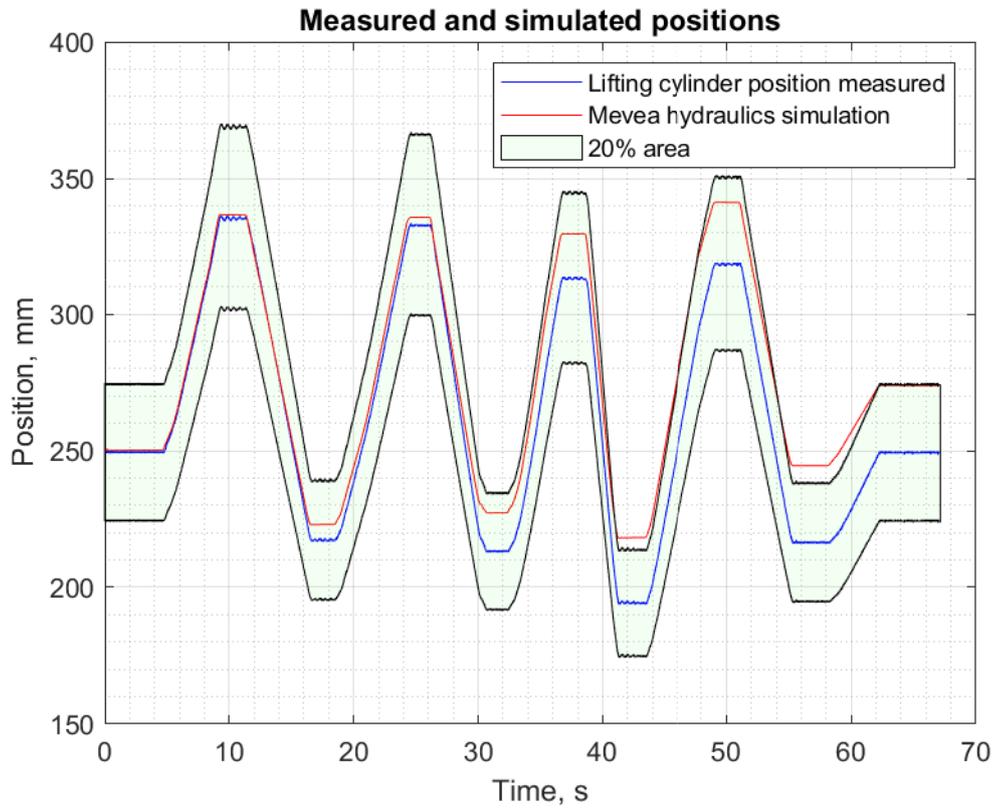


Figure 31. Lift cylinder positions for Mevea hydraulics (red) and measured data (blue).

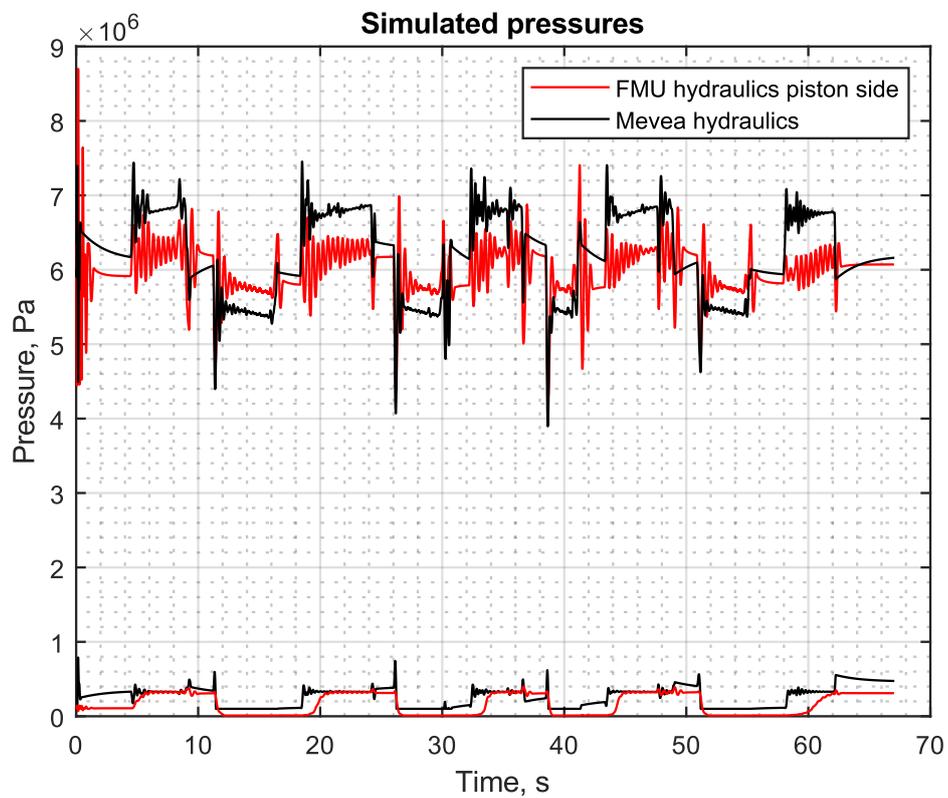


Figure 32. Simulated pressures for FMU (red) and Mevea hydraulics (black).

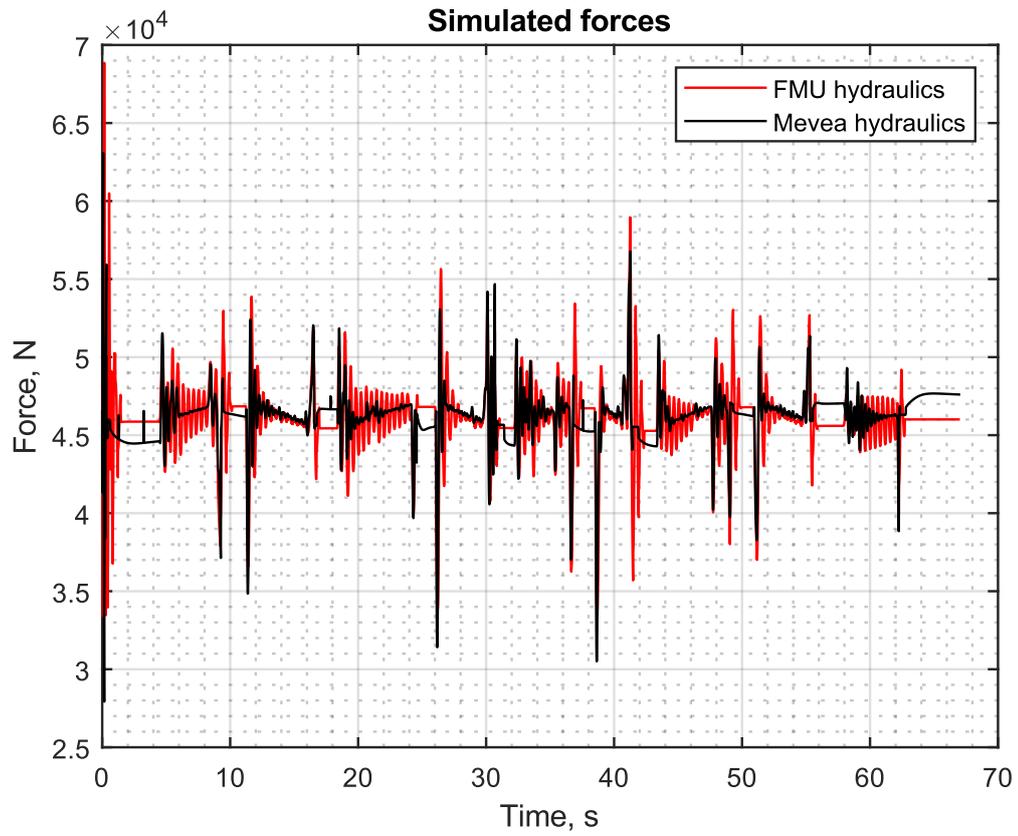


Figure 33. Simulated cylinder forces for FMU (red) and Mevea hydraulics (black).

Similar results for pressures and forces could be expected because Mevea uses equations 12 and 13 for calculating pressure in hydraulic chamber and equation 6 for cylinder force. However, instead of equation 7, Mevea calculates friction force based on cylinder force, cylinder co-efficient and velocity dependent friction co-efficient (Reference Manual for Solver Library, p. 77). The difference in position compared to measured data could be caused by same reasons as for the FMU model because both models use the same exact model for mechanics and by the small differences in used equations.

Figure 34 represents the loop duration for the simulation with FMU hydraulics and model with Mevea's hydraulics. Based on loop duration, using FMU for this model doesn't affect the model's simulation time at all because both simulations are ready with calculating 1 ms timestep in less than 0.3 ms and the largest difference between the simulations is 0.02 ms.

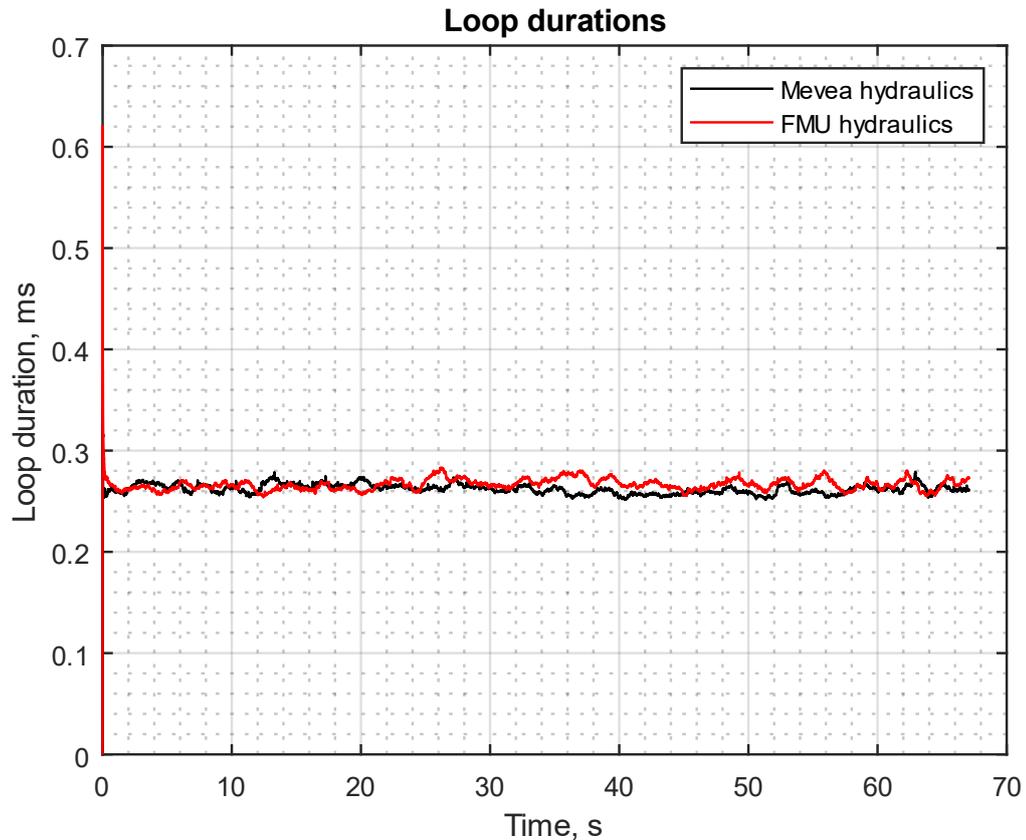


Figure 34. Loop durations for FMU simulation (red) and Mevea hydraulics (black).

Based on the results from this chapter, one advantage gained by building the hydraulic circuit with Simulink is that all the used equations and values are known. With the FMU hydraulics the user can see how all the calculations are done and which parameters are used but with the hydraulic circuit made with Mevea the user inputs values and the ready-made component then outputs new values and the user doesn't know how the output is achieved. However, Mevea does introduce the used equations in the Reference Manual for Solver Library, so if needed, the user does know how the calculations are done, but they aren't represented directly in the software. Second advantage in building hydraulic circuit with Simulink is customization. Customization of equations is beneficial for example in cases where uncommon phenomenon affects the system more than the regular activity, or in cases where the components are modeled in different ways et cetera, for example in this case, the friction force.

4.3 Key findings

Using FMU turned out to be very simple and good solution with Simulink and Mevea and also had good results with both simulation results and simulation speed. The largest problem with the FMU during this thesis was the fact that MATLAB and Simulink is only available in 64-bit version and the used Mevea was 32-bit version, and the problem is that with FMI for Co-Simulation with generated code requires a .dll file and it must be in the same format as the master program. Since the Mevea was 32-bit version, the FMU had to be exported with a 32-bit .dll, and for that Simulink requires Microsoft Visual C++ compiler. Even though in this thesis it was only an extra step, but for other use it means that the target operating system and processor must be known. However, according to FMI standard, this problem could be eliminated by using binary FMU (Modelica Association 2020, p. 11).

Based on the results in this thesis, FMU, especially FMI for Co-Simulation with generated code is great option for simulating systems which requires using multiple different software as long as the software support FMU import and export. Currently FMI 2.0 supports over 100 tools and the standard is constantly being developed by multiple companies and research institutes, which means that it is a solid option for building systems which require multiple components and subsystems (Modelica Association 2020, p. 1).

4.4 Topics for future research

If more accurate simulation results are needed in the future, the Patu-655 model could be developed by studying the 4-bar mechanism more because the model accuracy is very sensitive to it and for example this could include adjusting the constraints in the Mevea model. Also the hydraulic circuit could be developed further for example by adding leakage to the cylinder equations. In addition to these, also control system could be developed for the model instead of using joystick data as input. The control system could be imported as a separate FMU or with the Transmission Control Protocol/Internet Protocol socket interface method.

5 CONCLUSION

The aim of this thesis was to model Patu-655 loader's mechanics with Mevea and hydraulics with Simulink and combine these two simulation models with FMI. Mevea was chosen as software for the mechanics because it is capable of real-time simulation of complex systems. Simulink was chosen for the hydraulics because it has a simple and clear interface for mathematical modeling, and it introduced "Export for co-simulation standalone FMU" in version R2020a, and the interaction between Mevea and Simulink hasn't been studied before in LUT University with FMI for Co-Simulation with generated code technique. Previously the interface has been achieved with tool coupling method, which requires both simulation software to be open, which can make the simulation computationally more demanding, but with generated code method only the master software must be running.

The research started by building hydraulic circuit in Simulink with mathematical equations. The hydraulic circuit was verified by connecting the circuit to Simscape Multibody model that was used in Shevchuk (2020) study so the circuit could be observed without Mevea affecting the results. After that, the multibody model was built in Mevea with Patu-655 models by Luostarinen et al. (2014). The functionality of multibody model was tested by building simple hydraulic circuit with Mevea's in-built tools. After both models were tested independently, the simulation models were combined with FMI. The interface was achieved by using FMI for Co-Simulation with generated code method, where Mevea acts as master software and Simulink as slave. The final FMU had inputs for cylinder positions and velocities and outputs for cylinder forces.

The model built in this thesis was compared to simulation results and measured data from Shevchuk (2020) master's thesis because both systems used same hydraulic circuit, but Shevchuk (2020) used Simscape Multibody for mechanics. Both systems were simulated with Runge-Kutta 4 solver and with 1 ms timestep but the FMU had 0.25 ms timestep for the hydraulic circuit. Models were simulated with 140 bar pump pressure and with same parameters, but the 3D-model used in this thesis had some minor differences compared to the real-life system. For example the 4-bar mechanism is simplified and instead of a mass at

the end of the tilt boom, the 3D-model has a log grapple which acts as a mass in the simulations.

The model developed in this thesis achieved similar results to Shevchuk (2020) simulations and measured data, but it did have some differences. The FMU simulation model behaved similarly but achieved lower values with pressures and forces, but the cylinder position did stay inside 10 % error margin when compared to measured data from laboratory tests. However, the differences were small enough and the model can be considered working correctly. The results were verified by multiple simulations with different control signals, pump pressures and starting positions.

Largest difference between the FMU simulation model and Shevchuk (2020) model was simulation time. The model developed in this study was able to run the simulation in real time, while Shevchuk (2020) simulation with Simscape Multibody and Simulink for the same model took 15 minutes to run 100 second simulation. The drastic difference could be explained by difference in computing power, using different software or solution method for the model's dynamics. The effect of different multibody formalism was tested with the FMU model. The initial model used recursive formulation which is computationally more efficient than Lagrangian method. With the recursive formulation it took less than 0.3 ms for calculating one timestep, while with Lagrangian formulation it was constantly over 0.5 ms.

The hydraulic circuit was also built with Mevea's own tools for hydraulics so that the difference between custom built hydraulic circuit and Mevea's hydraulic could be compared. The model with Mevea's hydraulics achieved similar results to the FMU model, but the cylinder position didn't stay inside the 10 % error margin when compared to measured data. The difference between the FMU model and Mevea's hydraulics could be explained by small differences in equations. Mevea for example calculates bulk modulus and friction force differently than the custom built hydraulic circuit. Most notable finding from building the hydraulics with Mevea was that the loop duration was same for both models, meaning that using FMU for this model didn't affect the model's simulation time at all.

Based on the results from this thesis FMI for Co-Simulation with generated code is a valid option for system simulations, where multiple software are needed. The FMU model had

good results with both simulation results and simulation speed, but it also provides other benefits. For example by building a custom hydraulic circuit, the user can see how all the calculations are achieved and which parameters are being used, while with ready-made components you can't see them directly and in this case the user would have to search the equations from Mevea's documents. Another important benefit for using custom hydraulic circuit was customization of equations. The customization is beneficial in cases where some phenomenon affects the system more than the regular activity, for example in this case the bulk modulus was calculated based on the amount of air in the hydraulic liquid because it was considered to have more impact than hydraulic hoses. However, the FMI for Co-Simulation with generated code had one drawback, it generates a dynamic linker which is platform dependent, meaning that the operating system must be known, but this problem could be avoided by using binary FMU.

In future research, the 4-bar mechanism of Patu-655 could be modeled more closely because the model is very sensitive to the 4-bar mechanism accuracy. Also a control system could be developed and imported to Mevea as separate FMU or with Transmission Control Protocol/Internet Protocol socket interface. The hydraulic circuit could also be built with different software and imported to Mevea as FMU. This would allow studying how using multiple FMUs in same model affect the simulation speed and results.

LIST OF REFERENCES

- Abel, A., Blochwitz, T., Eichberger, A., Hamann, P.M., & Rein, U. (2012). Functional Mock-up Interface in Mechatronic Gearshift Simulation for Commercial Vehicles. Proceedings of the 9th International Modelica Conference. Pp. 775-780.
- Andersson, S., Söderberg, A., & Björklund, S. (2007). Friction models for sliding dry, boundary and mixed lubricated contacts. Tribology international, 40(4), Pp. 580-587.
- Blochitz, T., Otter, M., Åkesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., Olsson, H. & Viel, A. (2012). Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. Proceedings of the 9th International Modelica Conference. Pp. 173-184.
- Chrisofakis, E., Junghanns, A., Kehrer, C. & Rink, A. (2011). Simulation-based development of automotive control software with Modelica. Proceedings 8th Modelica Conference. Pp. 1-7.
- Functional Mock-up Interface. (2021). [Functional Mock-up Interface webpage]. [Referred 1.11.2021]. Available: <https://fmi-standard.org/>
- Gomes, C., Thule, C., Broman, D., Larsen, P. G., & Vangheluwe, H. (2018). Co-Simulation: A Survey. ACM Computing Surveys, 51(3), Pp. 1–33.
- Gupta, M. (2021). Simulation of vertical people transportation systems. Master's thesis. LUT University. 89 p.
- Kaupila H. (2019). Lentokoneen päälaskutelineen yhteissimulaatio. Master's thesis. LUT University. 68 p.
- Khadim, Q. (2017). Real time simulation of the counterbalance forklifter with Mevea software. Master's thesis. LUT University 53 p.

Khaled, A. B., Gaid, M. B., Simon, D., & Font, G. (2012). Multicore simulation of powertrains using weakly synchronized model partitioning. *IFAC Proceedings Volumes*, 45(30), Pp. 448–455.

Kiani-Oshtorjani, M. (2020). *Real-Time Efficient Computational Approaches for Hydraulic Components and Particulate Energy Systems*. Doctoral thesis. LUT University. 83 p.

Luostarinen, L., Åman, R & Handroos, H. (2014). Development of control interface for HIL simulation of electro-hydraulic energy converter. *Int. Rev. Modelling and Simulations (IREMOS)*, 7(4), Pp. 653-660.

Mäkitalo, J. (2020). *System Simulation of forest machines*. Master's thesis. Tampere University. 44 p.

Malysheva, I., Handroos, H., Zhidchenko, V. & Kovartsev, A. (2018). Faster than real-time simulation of a hydraulically actuated log crane, 2018 Global Fluid Power Society PhD Symposium (GFPS), IEEE, Pp. 1–6.

Malysheva, J. (2021). *Faster than real-time simulation of fluid power-driven mechatronic machines*. Doctoral thesis. LUT University. 79 p.

MathWorks. (2021a). *FMI Target for Simulink Coder*. [MathWorks webpage]. [Referred 6.11.2021]. Available: https://se.mathworks.com/products/connections/product_detail/iti-fmi-target.html

MathWorks. (2021b). *exportToFMU2CS*. [MathWorks webpage]. [Referred 17.11.2021]. Available: <https://se.mathworks.com/help/slcompiler/ref/exporttofmu2cs.html>

Mevea Ltd. *Mevea Modeller: Beginner tutorials*. Version 2.0.

Mevea Ltd. *Reference Manual for Solver Library 7.70*. Version 7.70.2858.

Modelica Association. (2020) Functional Mock-up Interface for Model Exchange and Co-Simulation. Version 2.0.2. 130 p.

Peiret, A., González, F., Kövecses, J., & Teichmann, M. (2018). Multibody system dynamics interface modelling for stable multirate co-simulation of multiphysics systems. *Mechanism and Machine Theory*, 127, Pp. 52–72.

Rahikainen, J., Mikkola, A., Sopanen, J., & Gerstmayr, J. (2018). Combined semi-recursive formulation and lumped fluid method for monolithic simulation of multibody and hydraulic dynamics. *Multibody System Dynamics*, 44, Pp. 293-311.

Rouvinen, A. Mevea Ltd. Mevea co-simulation via FMI. Training material. 17 p.

Shevchuk, D. (2020). Electrical-hydraulic converter simulation and testing. Master's thesis. LUT University. 53 p.

Ustinov, S. (2018). Development of a Novel Real-Time Model of Human Skeleton in Mevea Platform. Master's thesis. LUT University. 84 p.

I. Additional simulation results

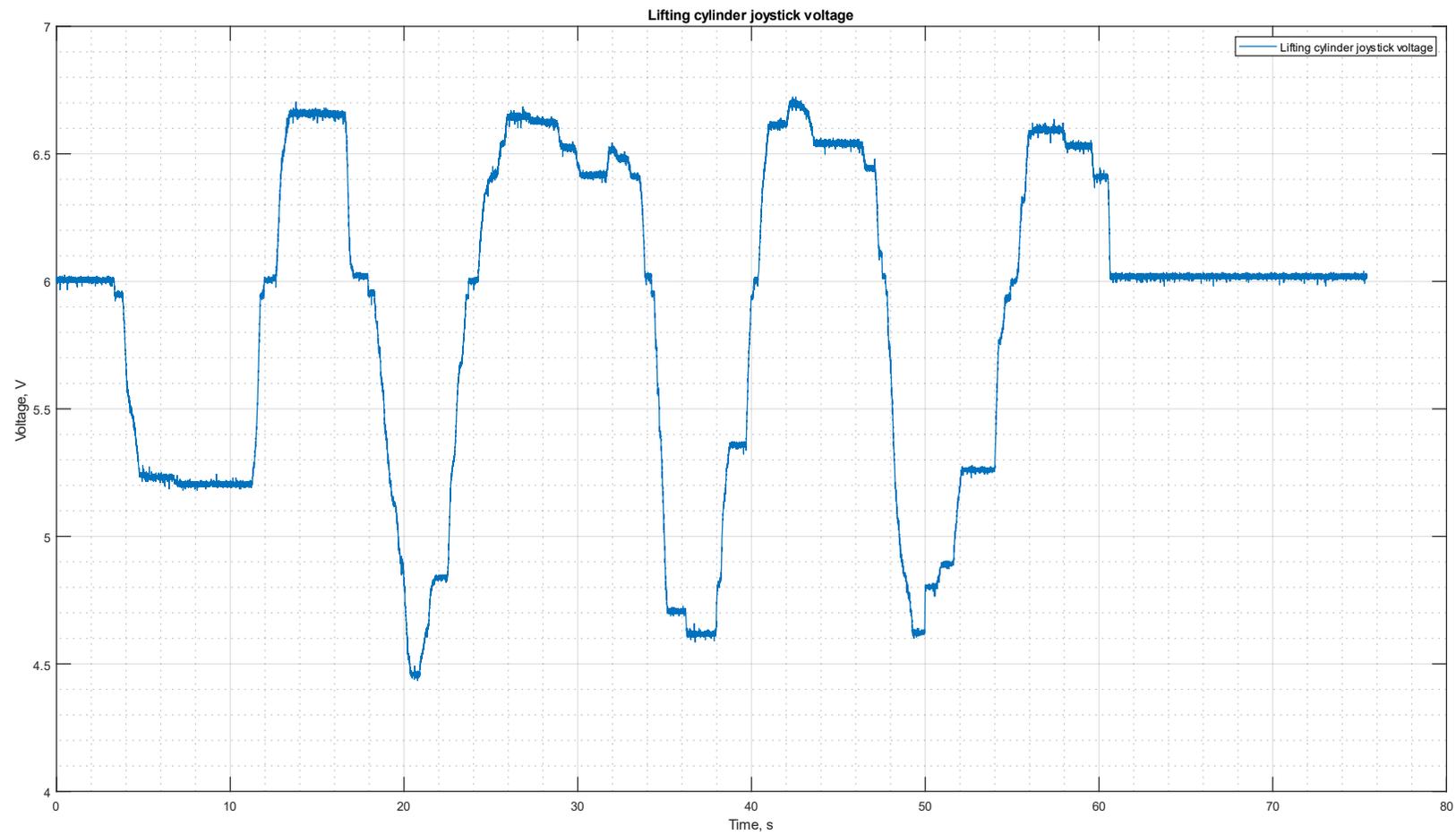


Figure I.1. Lifting cylinder input voltage for 80 bar pump pressure simulation.

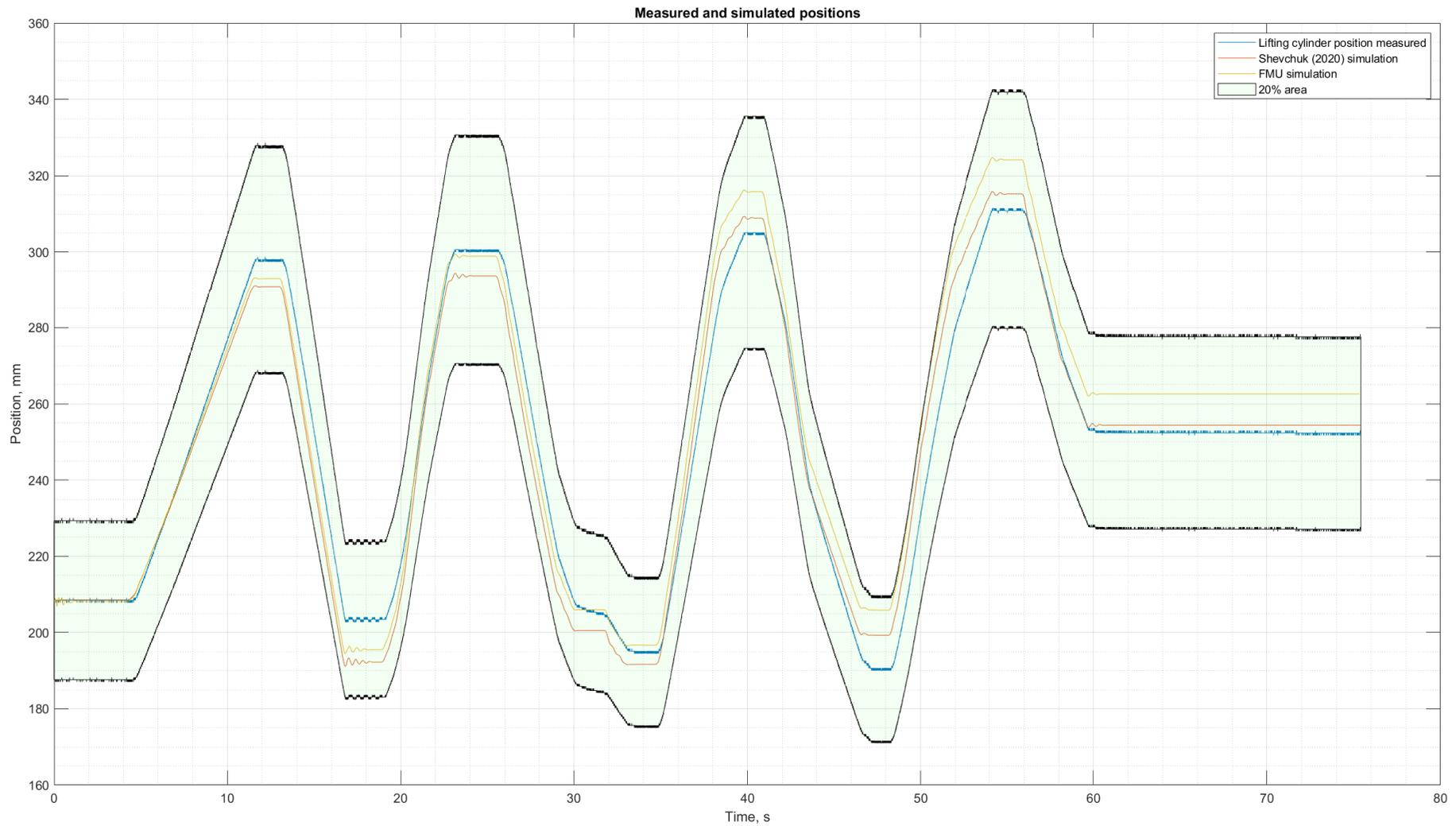


Figure I.2. Lifting cylinder measured and simulated positions for 80 bar pump pressure simulation.

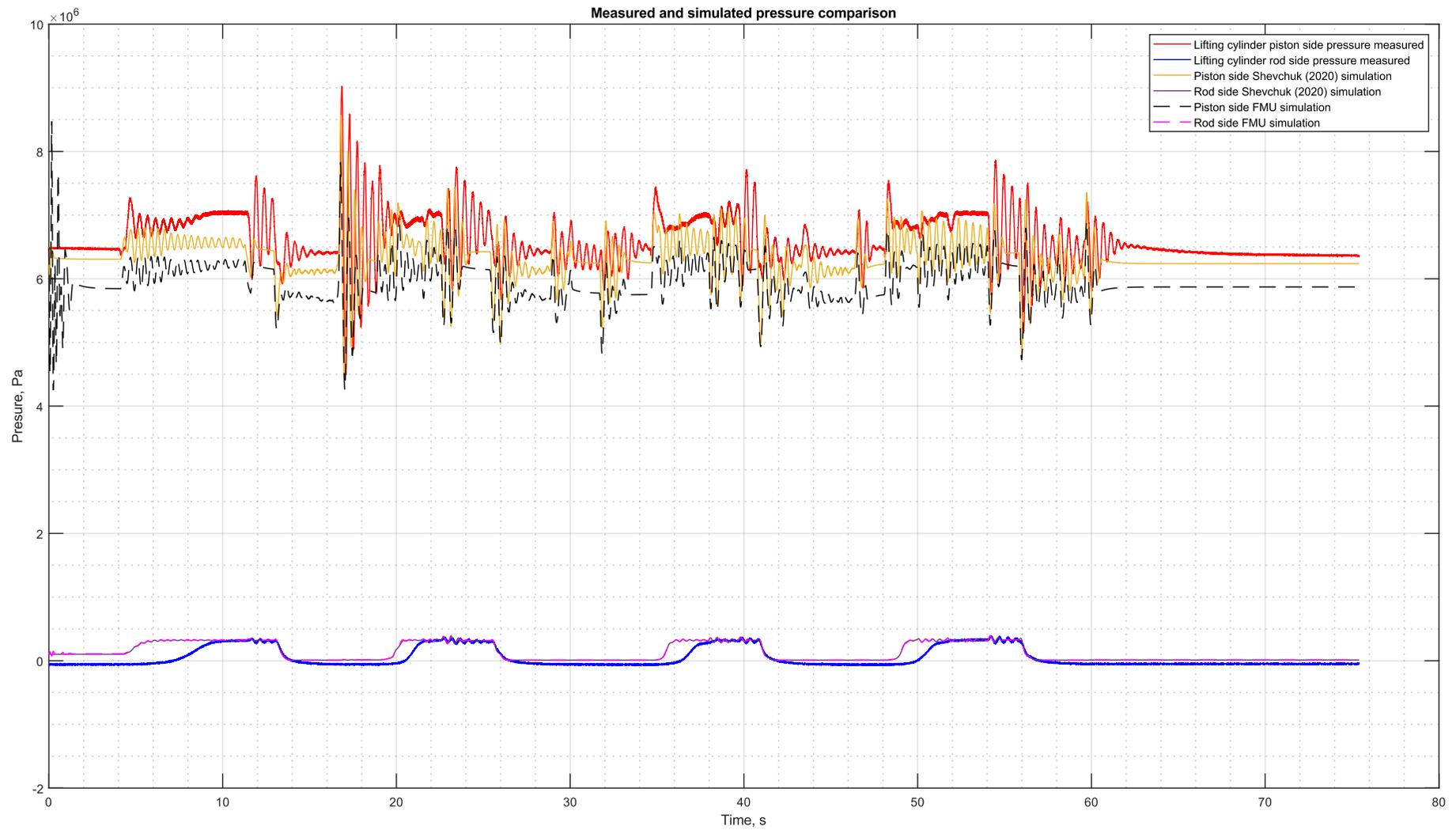


Figure I.3. Lifting cylinder measured and simulated pressures for 80 bar pump pressure simulation.

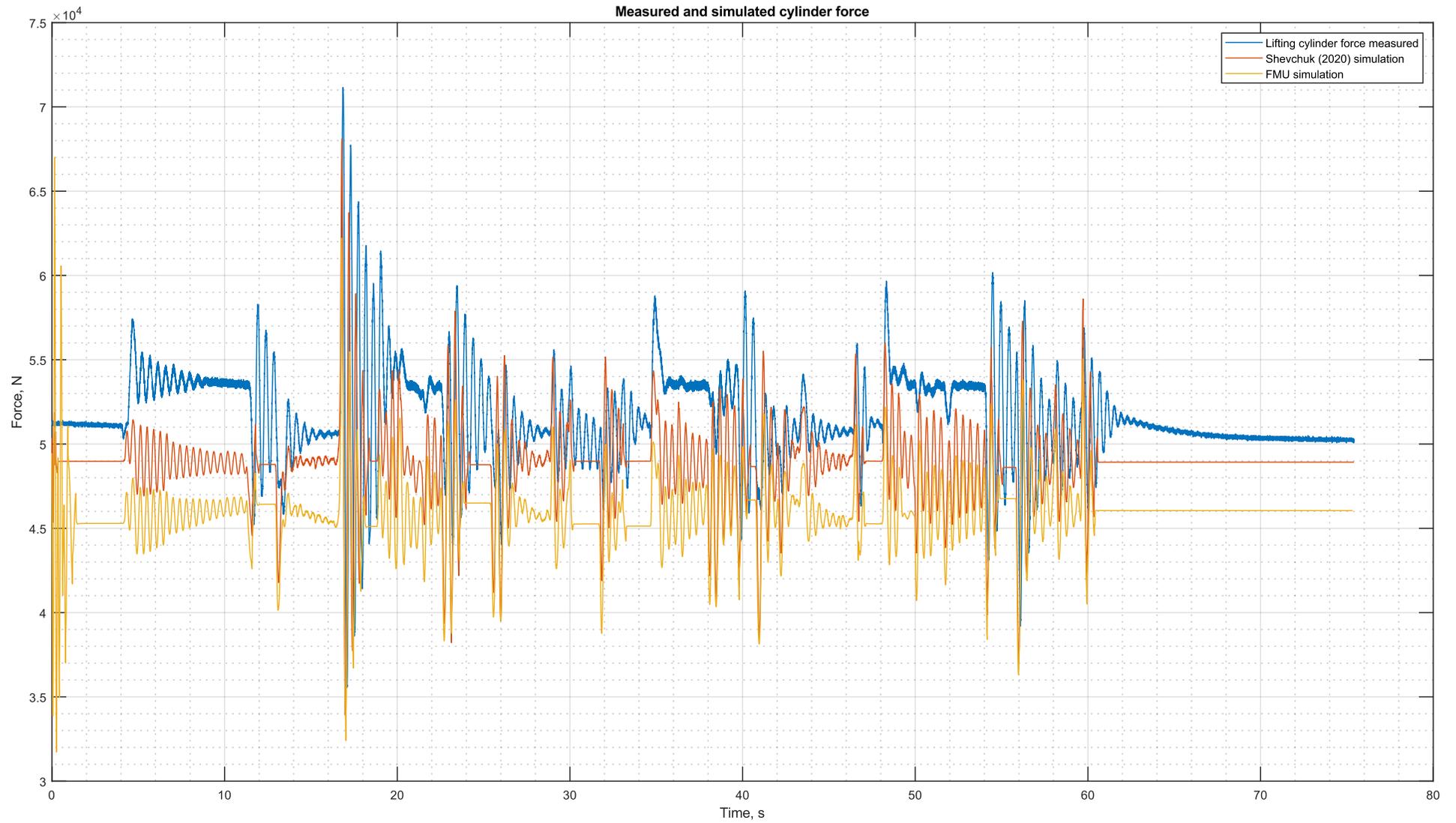


Figure I.4. Lifting cylinder measured and simulated forces for 80 bar pump pressure simulation.

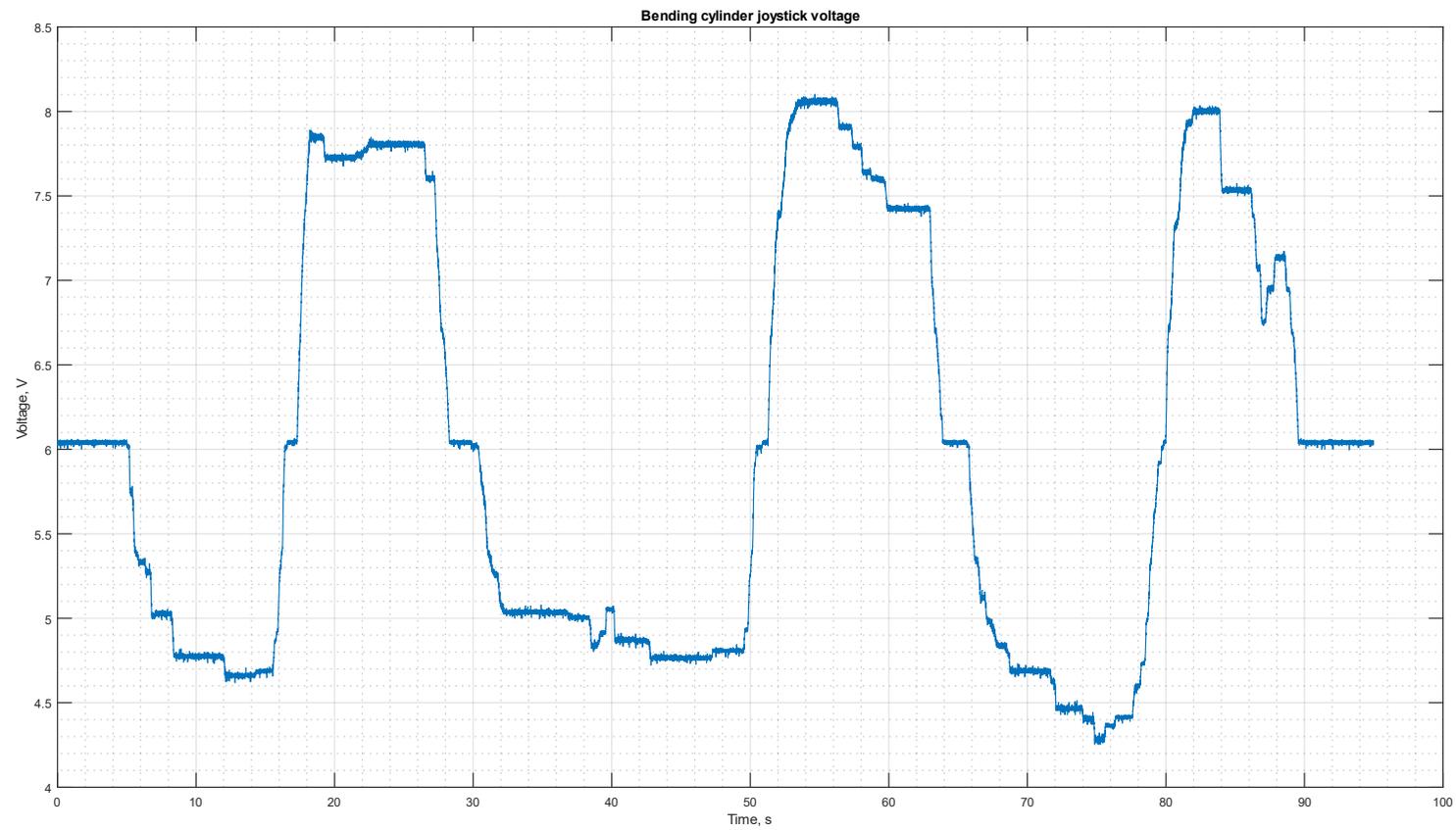


Figure I.5. Tilt cylinder input voltage for 100 bar pump pressure simulation.

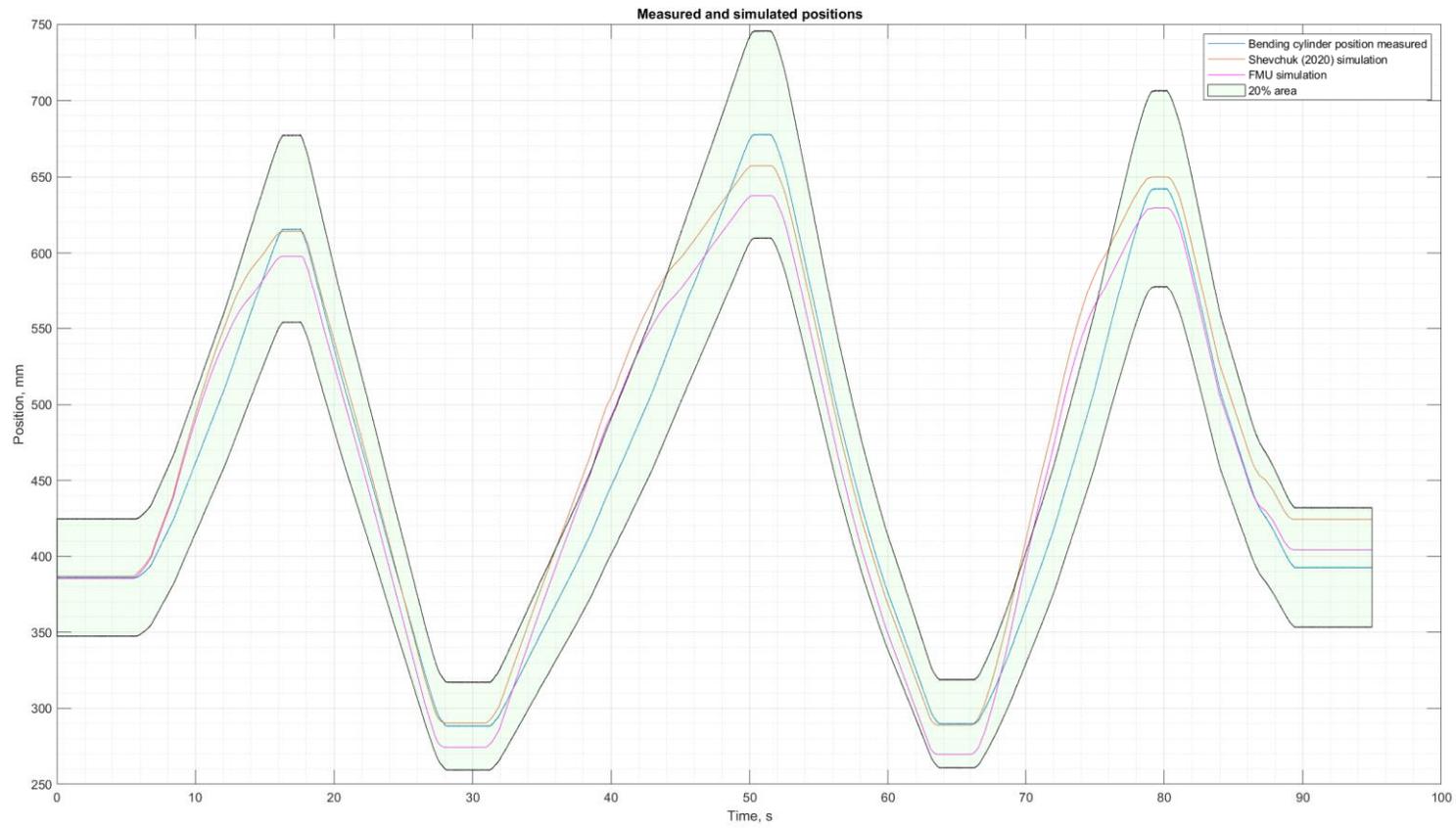


Figure I.6. Tilt cylinder measured and simulated position for 100 bar pump pressure simulation.

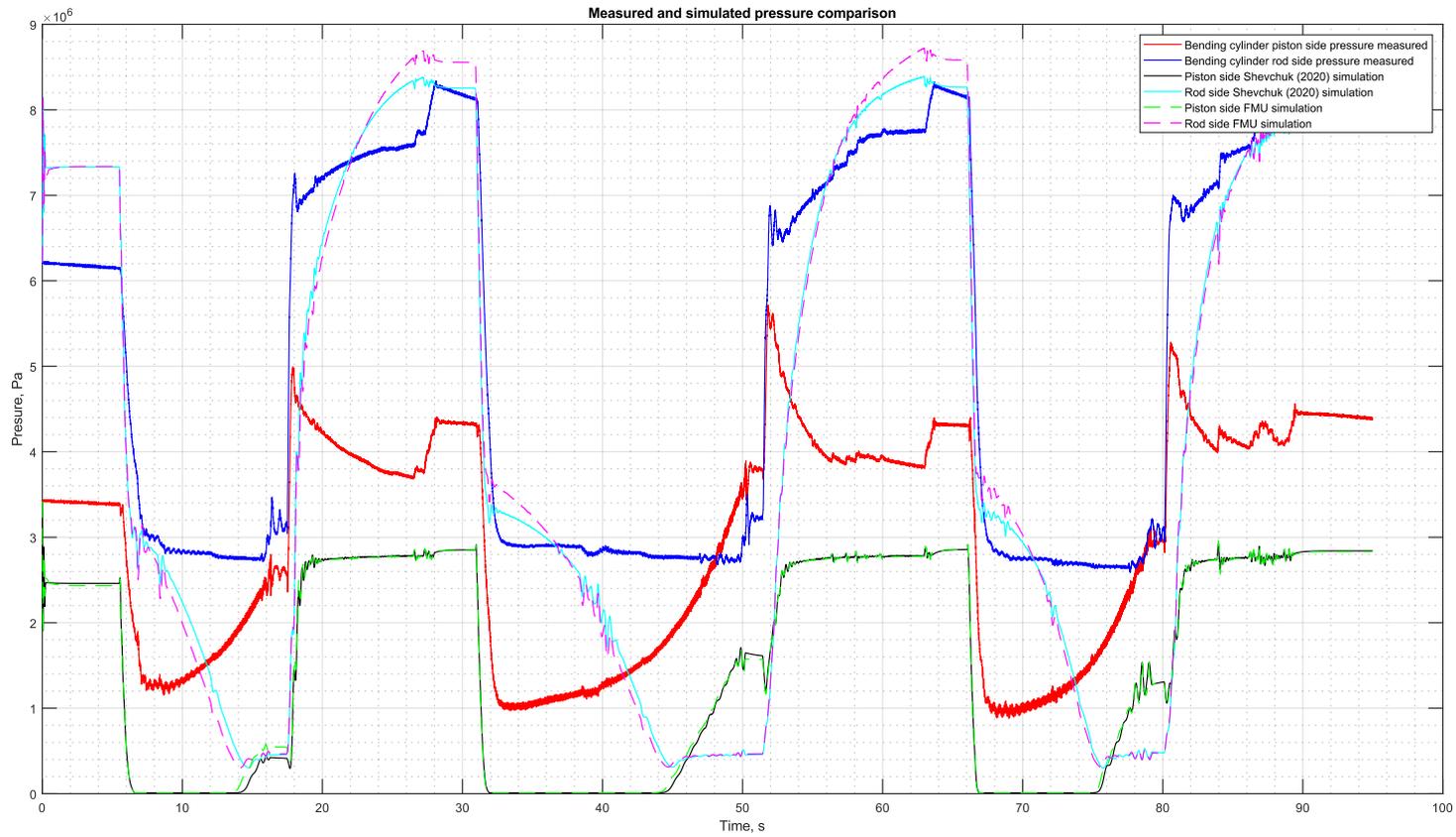


Figure I.7. Tilt cylinder measured and simulated pressures for 100 bar pump pressure simulation.

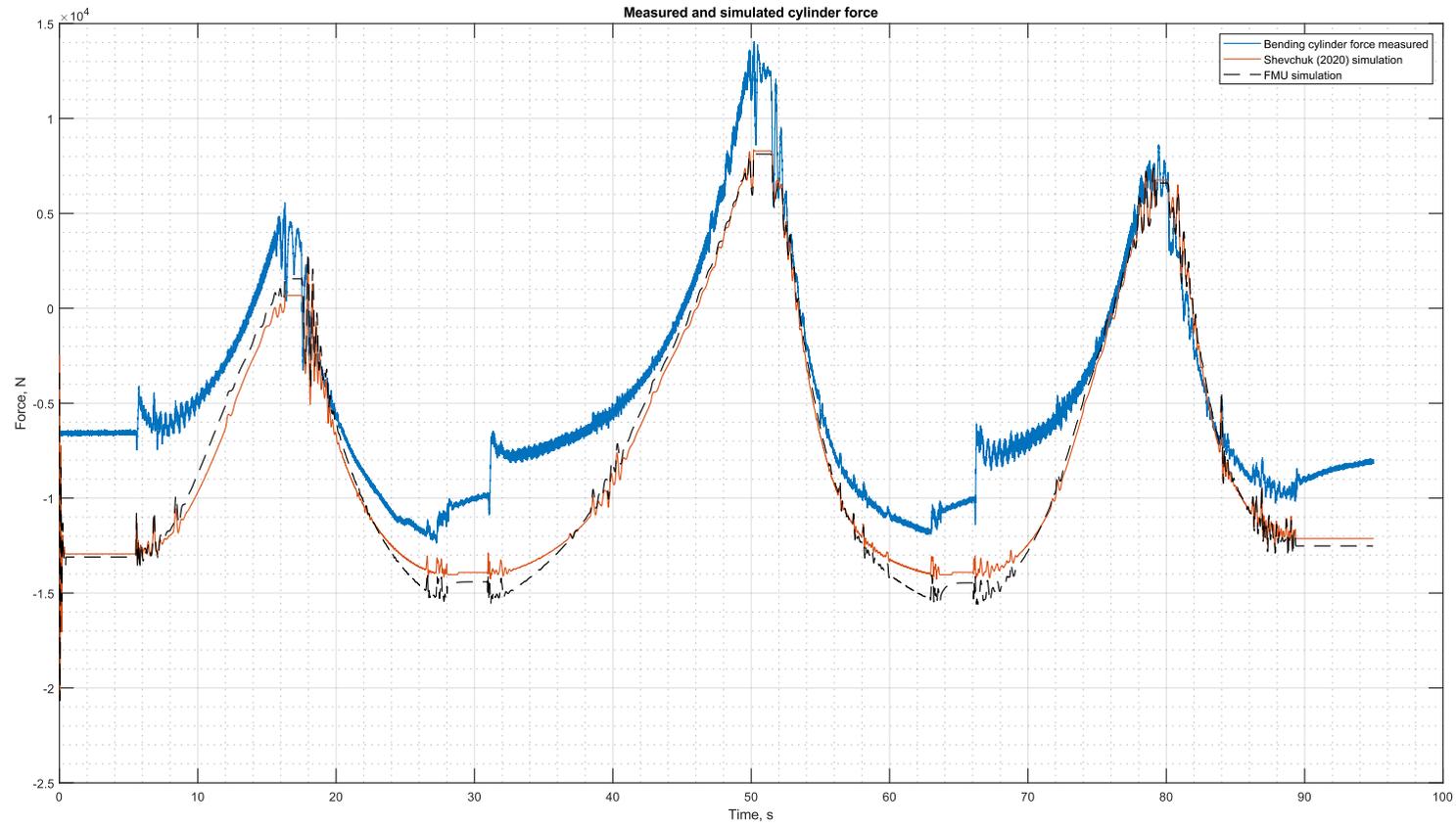


Figure I.8. Tilt cylinder measured and simulated forces for 100 bar pump pressure simulation.

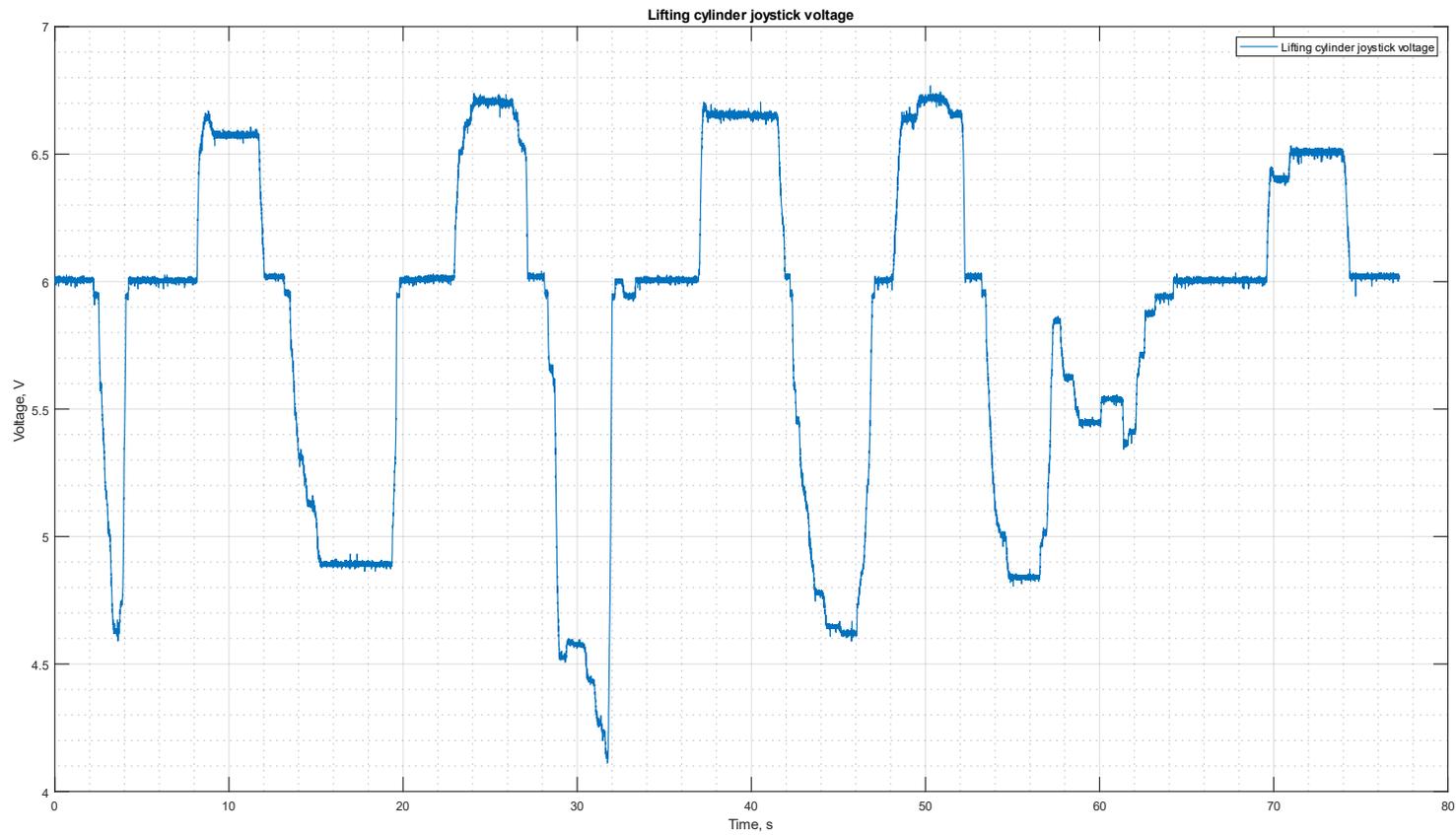


Figure I.9. Lift cylinder input signal for 120 bar pump pressure simulation.

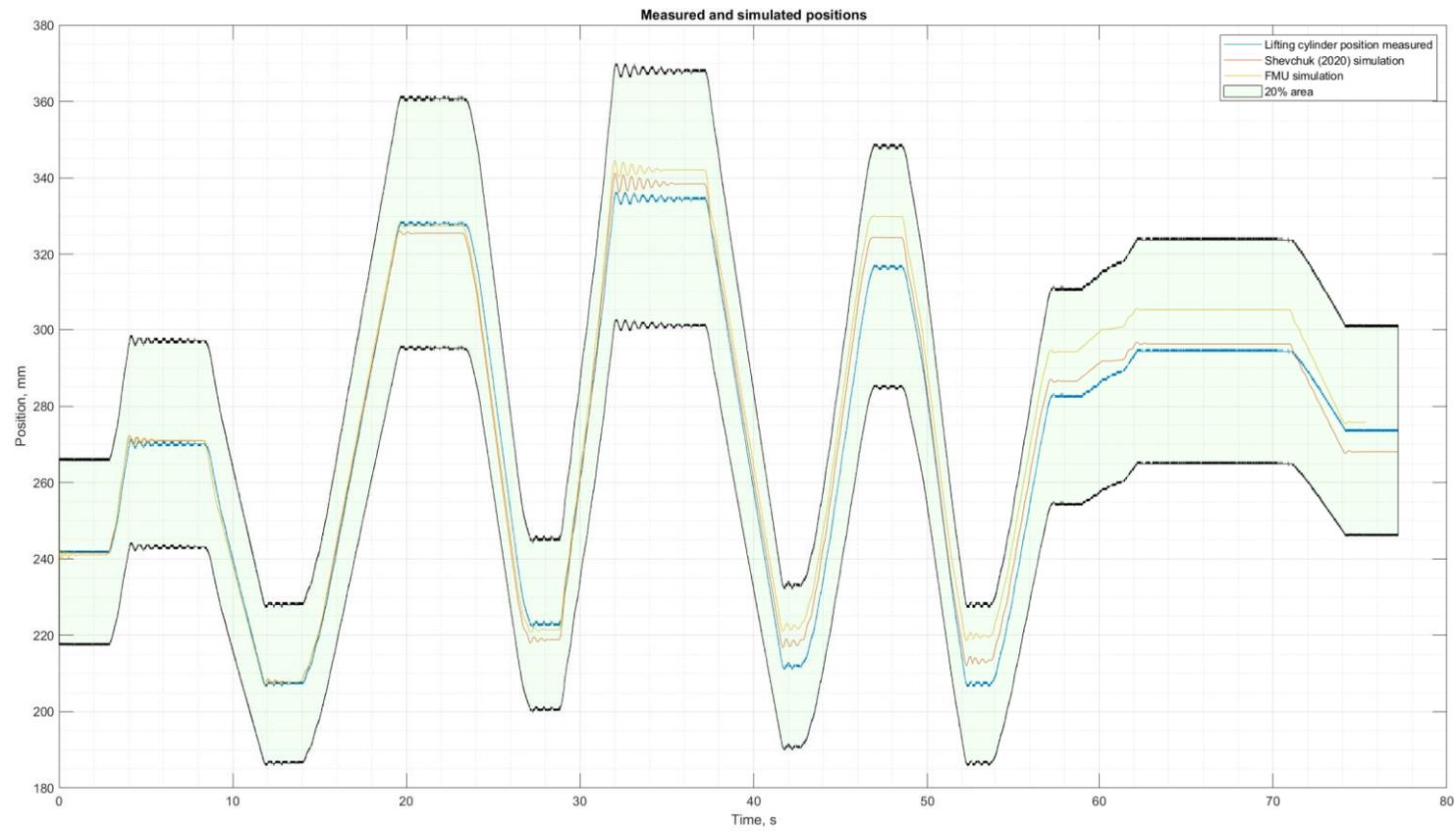


Figure I.10. Lift cylinder measured and simulated positions for 120 bar pump pressure simulation.

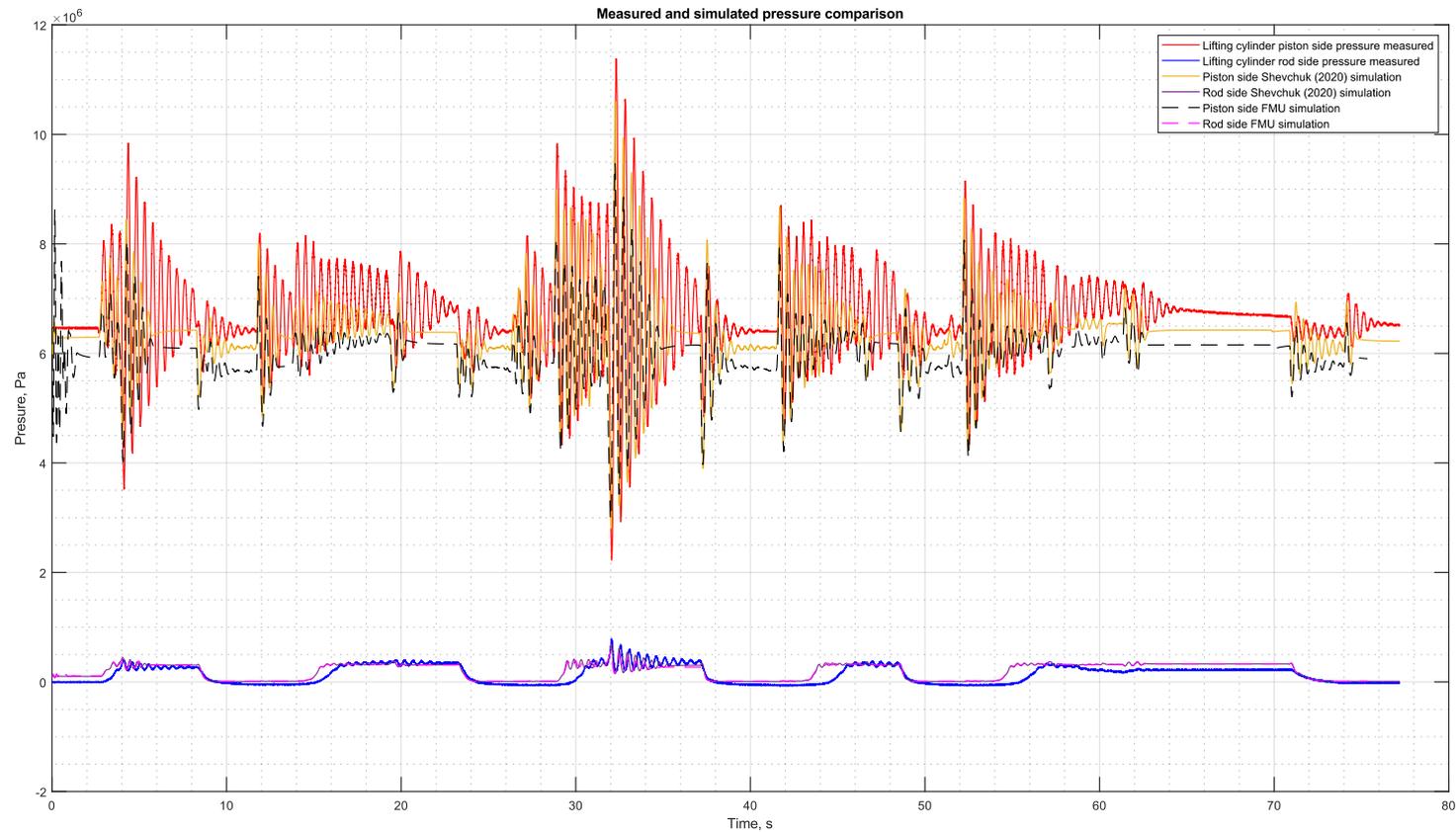


Figure I.11. Lift cylinder measured and simulated pressures for 120 bar pump pressure simulation.

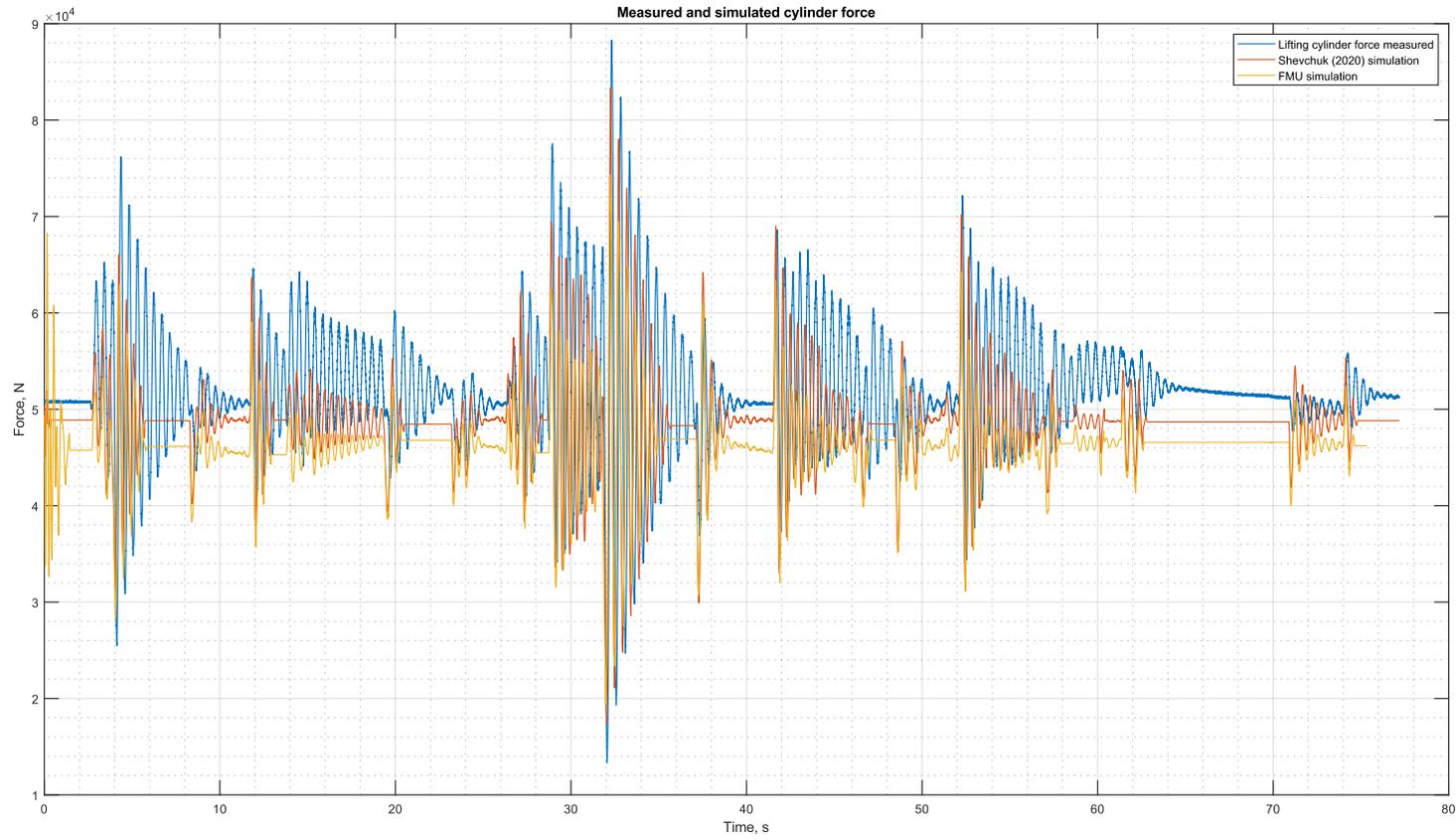


Figure I.12. Lift cylinder measured and simulated forces for 120 bar pump pressure simulation.

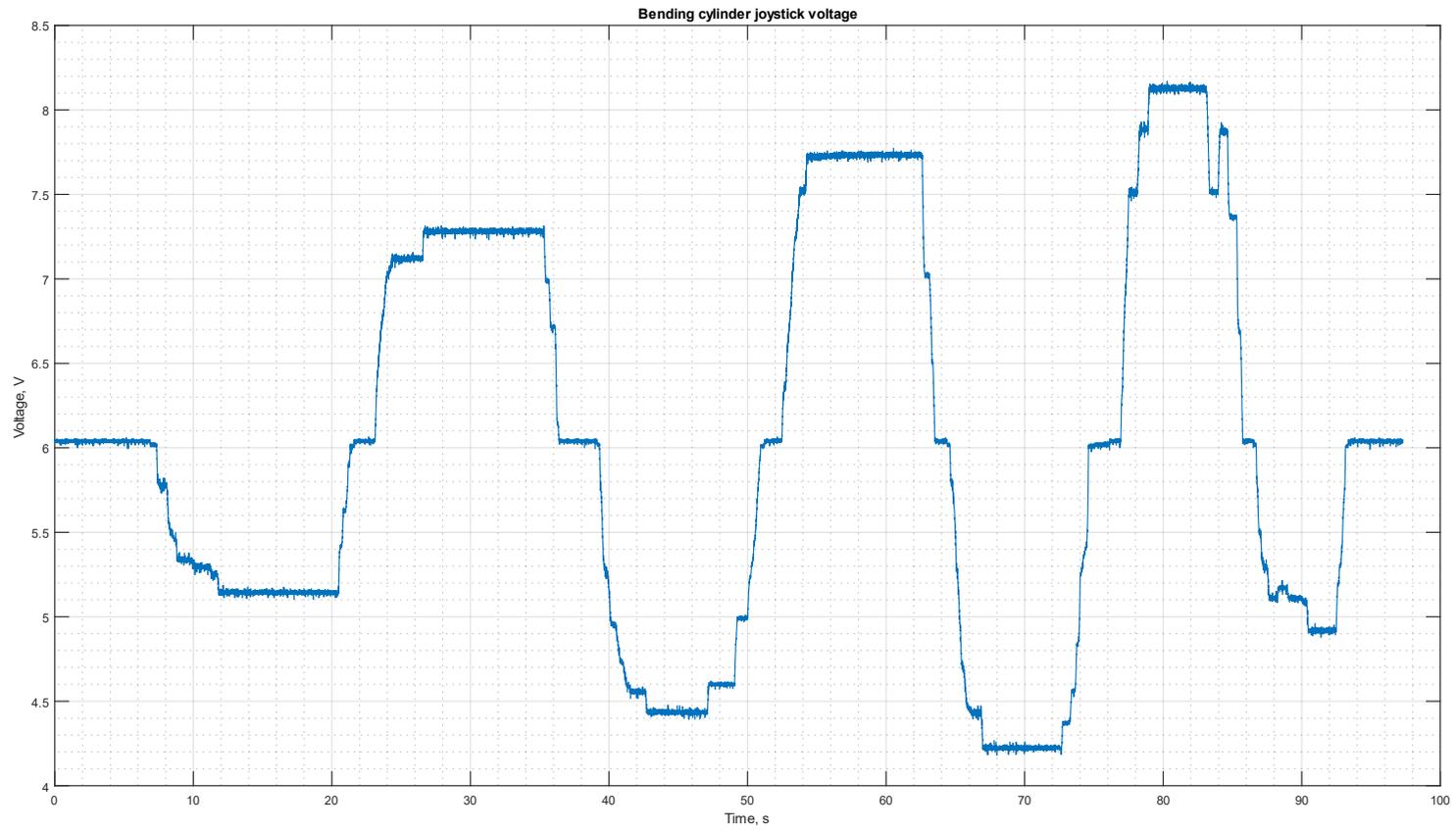


Figure I.13. Tilt cylinder input signal for 120 bar pump pressure simulation.

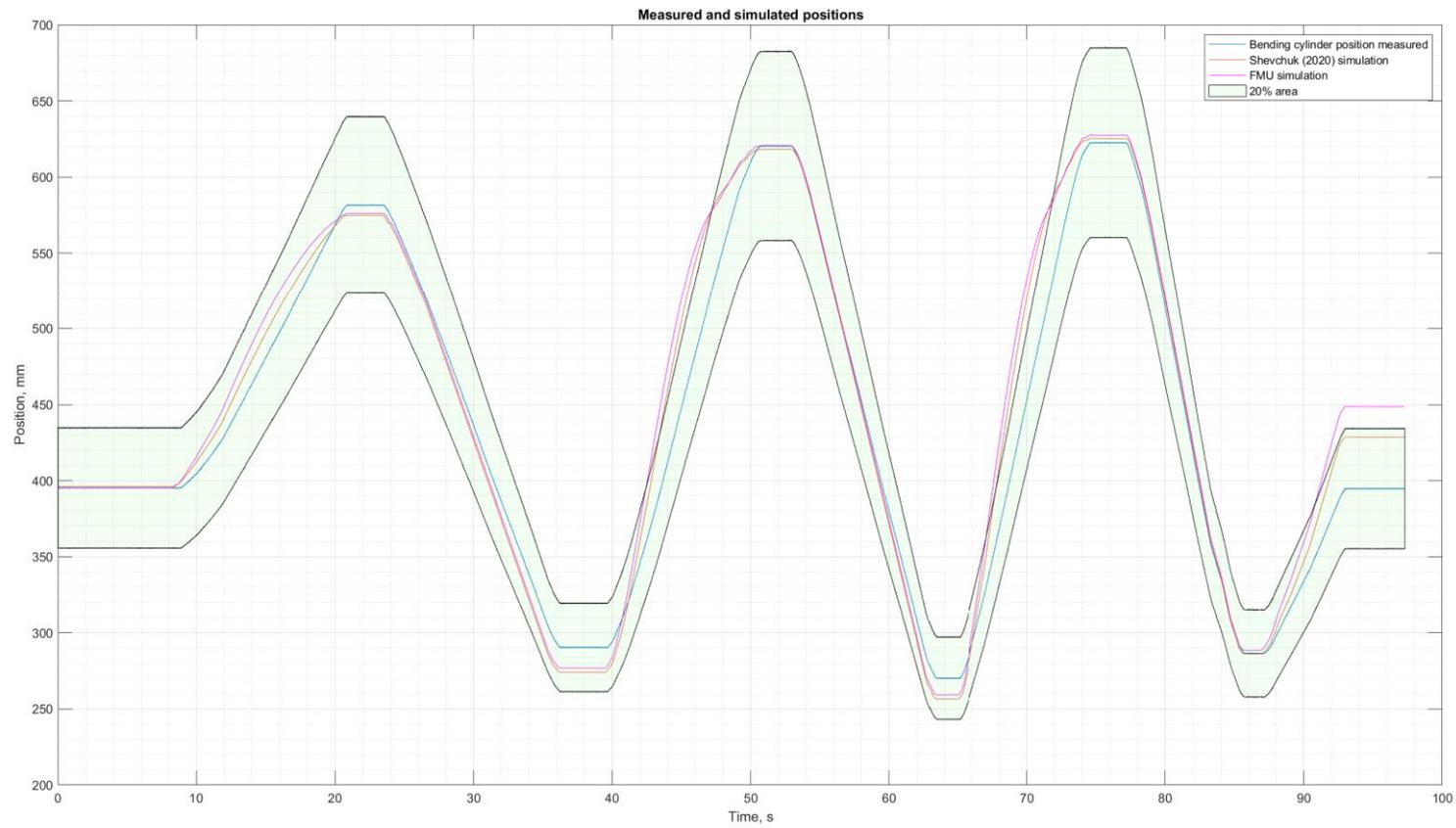


Figure I.14. Tilt cylinder measured and simulated positions for 120 bar pump pressure simulation.

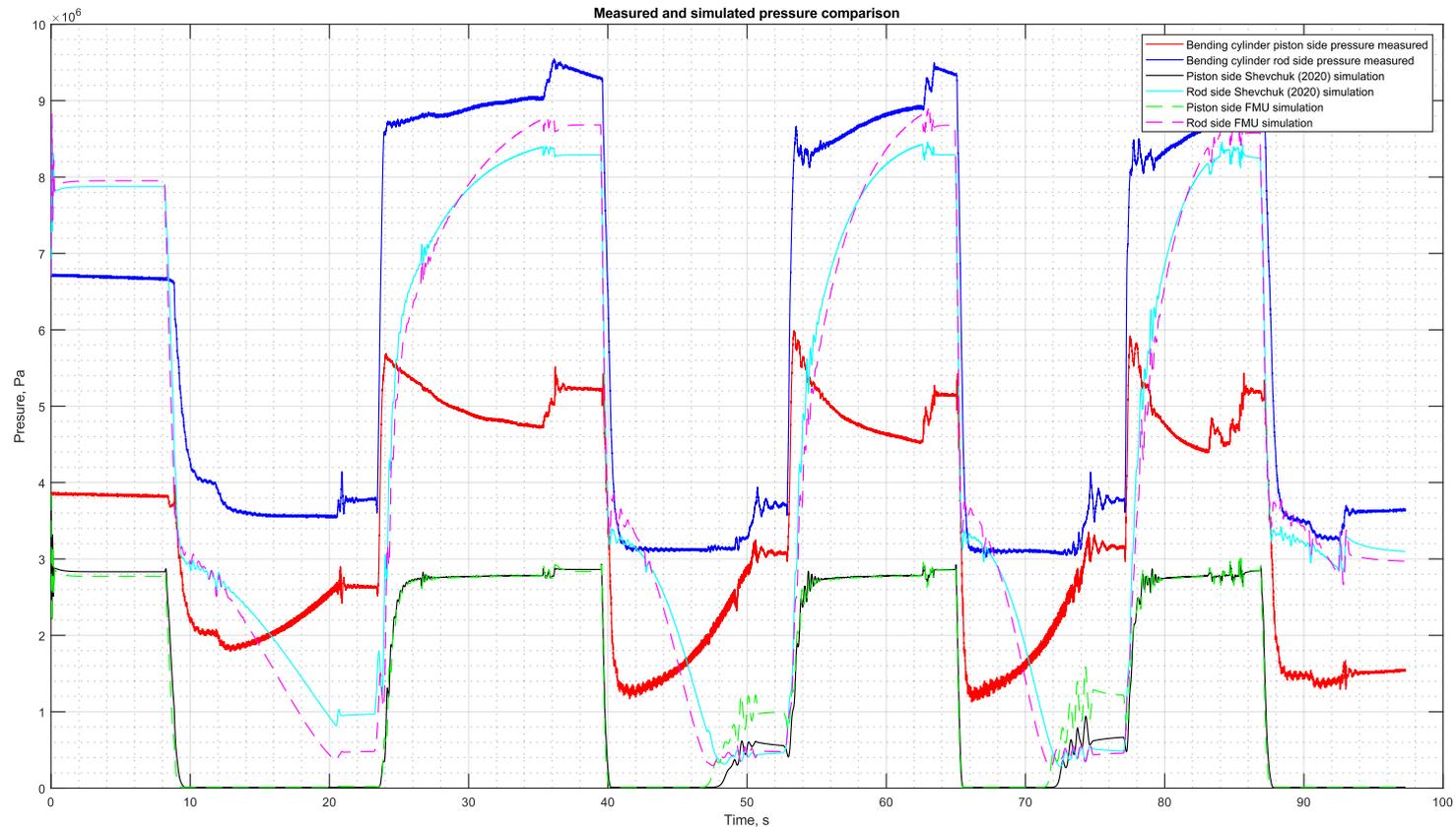


Figure I.15. Tilt cylinder measured and simulated pressures for 120 bar pump pressure simulation.

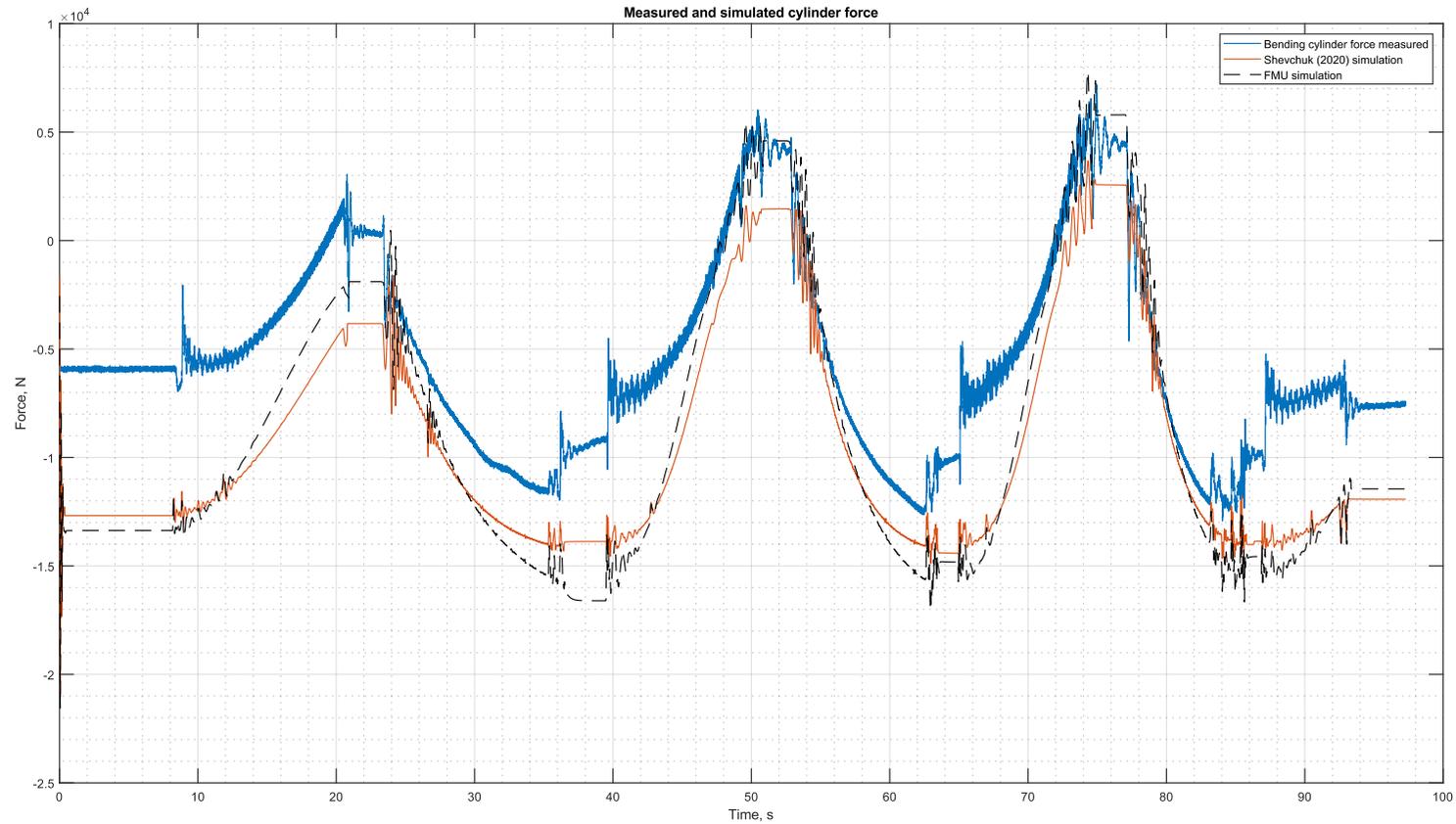


Figure I.16. Tilt cylinder measured and simulated forces for 120 bar pump pressure simulation.