Integration Project IEM

Investigating the dynamic behaviour of a multi-piston pump

Lennard Hut (S2718960)

Bachelor Integration Project performed at The Ocean Grazer Project

First supervisor : prof. dr. A. Vakis
Second supervisor : G.K.H. Larsen, PhD
Student : L.Y. Hut

25 June 2018

University of Groningen
Faculty of Science and Engineering
Industrial Engineering and Management programme
9747 AG Groningen

project
June 2018

Abstract

As the world's energy demand is becoming higher and fossil fuel reserves are starting to become less, more people are starting to think about green energy. The Ocean Grazer is a device that can compete with other methods to obtain renewable energy, as well as the methods for extracting fossil fuels. The device is designed in such a way that it is able to extract the energy of waves. Besides extracting wave energy, the current concept is also able to store wind energy. To extract the wave its energy, the Ocean Grazer makes use of a system which is called the Multi Pump Multi Piston Power Take-Off (MP^2TO) system.

Several research studies have already been performed concerning the pumping system of the Ocean Grazer. The present report builds on the past models of the single piston pump and measurements that were performed at the University of Groningen. This research that has already been done is used as a foundation to obtain a model which is able to predict the behaviour of the multi piston pump system.

By making use of a literature study and analysing the current model that is being used, the parameters of the other two piston subsystems, which have to be added, were determined. The parameters that were determined are added to the Matlab code, which supports the Simulink MPP model. The model that is designed is capable of simulating the behaviour of the MPP model and can provide the different piston combinations. The outputs that the model gave were used to analyse the simulated behaviour. The behaviour which is provided by the model can be verified when the MPP system is working in the experimental setup.

Keywords: Ocean Grazer, Multi Piston Pumping system, Simulink, Matlab, Simulation

Table of contents

1. Introduction

In 2015 the United Nations held a convention to discuss climate change with all of its members in Paris. During this convention, the world leaders agreed on taking action and the agreement of Paris was signed. The goals of the agreement are as the Paris agreement states: *"The aim is to strengthen the global response to the threat of climate change by keeping a global temperature rise this century well below 2 degrees Celsius above pre-industrial levels and to pursue efforts to limit the temperature increase even further to 1.5 degrees Celsius."* [1].

To obtain the goals of the Paris Agreement, certain types of fuels have to be discarded. The most polluting fuels are the fossil fuels. The polluting factor of these fossil fuels is the carbon that reacts with oxygen when the fuel is burnt. The carbon and oxygen will form CO_2 which is one of the greenhouse gasses that are causing the climate change. However, the fossil fuels are used to provide electricity for the world. The electricity demand of the world has only been increasing for the last few decades. In its annual Energy Outlook, the International Energy Agency outlook shows the change in energy demand and a prognose for the years to come. Figure 1[2] shows the electricity demand which was published in the Energy Outlook of 2017.

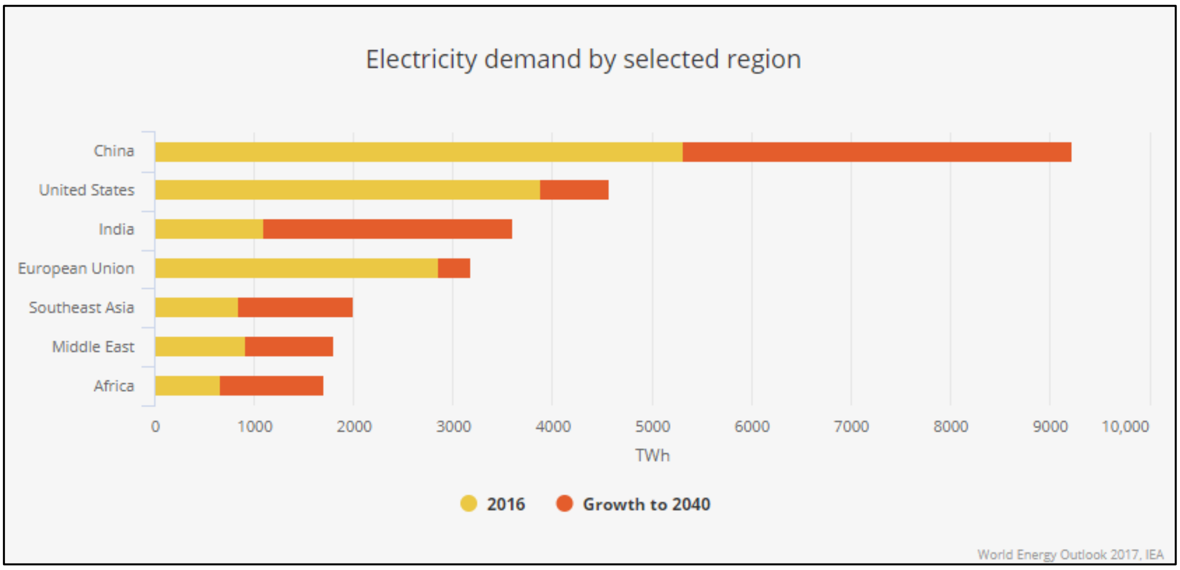


Figure 1: Electricity demand in the world[2]

The figure shows that the expected growth of the electricity demand is enormous in terms of TWh (terawatt hours). However, to be able to reach the goals of the Paris agreement, the use of the previously mentioned fossil fuels as energy sources must be reduced substantially.

However, during the last few decades, an enormous amount of research is done in finding new possible fuels or potential energy sources that will not harm the environment as much as the fossil fuels. These new types of environmentally friendly energy sources are called renewable energy sources. The big advantage of renewable energy sources is that they are not finite, i.e. they can be used again and again. Renewable sources that have been explored in the past decades and have become the mainstream sources are for example solar energy, wind energy or hydro power[3]. Besides research in the mainstream renewable energy sources, a large amount of research is done for emerging renewable energy sources. One of these emerging renewable resources is marine energy.

Marine energy is focused on the energy extraction which happens in the oceans. This type of energy resource does not necessarily mean that the water of the ocean has to be involved. As is currently done, large windfarms are placed in the oceans to capture energy. However, one can use the waves that are moving through oceans to extract energy as well. The availability of ocean waves is quite high since every ocean produces waves. However, the power of the waves deviates among the different oceans and different parts of the oceans. Figure 2[4] shows that the most powerful waves are concentrated in the Oceans in the southern hemisphere. However, there is one location in the Atlantic Ocean, west of Ireland, which also has waves with quite high power.

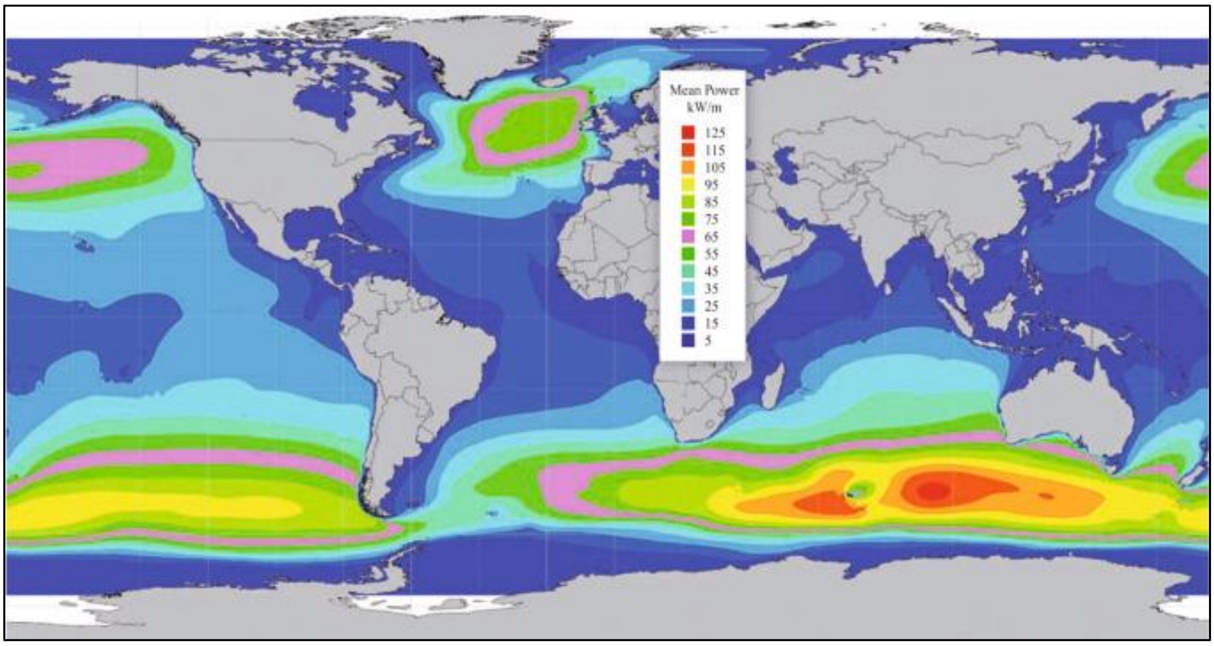


Figure 2: Mean wave power across the oceans[4]

Within Dutch universities, research is done on all kinds of renewable energy sources and eco-friendly inventions. The technical university of Delft won Elon Musk's hyperloop challenge[5] and, together with the technical universities of Eindhoven and Twente, it is competing in the world solar challenges. The University of Groningen (UG) is doing research in another direction. At the faculty of Science and Engineering, scientists are experimenting with a device which harvests energy from ocean waves. Yearly, research is done by a group of bachelor students, a group of master students and a group of academic staff members. The device that is being investigated is called the Ocean Grazer. The device can, when fully operational, be one of the solutions to meet the energy demand in renewable way.

Since 2011 research has been done in the Ocean Grazer. This paper will contribute to the knowledge about the Ocean Grazer device that already has been gathered. The Ocean Grazer, which will be discussed in depth in the next chapter, is currently still in the design phase. Little is known about the behaviour of one of the pumping systems, the multi-piston pump. In this research, the pumping subsystem of the Ocean Grazer will be investigated, where the focus lies on the force behaviour in the pumping motion.

2. Problem Analysis

2.1 Introduction

In this chapter, the problem analysis is described. The problem analysis is the first step of the research. This first step includes the context of the problem and the system, the description of the system, the problem statement, an analysis of the different problem owners, the stakeholders that are involved and the goal and research question(s) of the project.

2.2 Context

2.2.1 The Ocean Grazer project

Renewable energy is becoming more and more important in the energy consumption. Fossil fuel reserves are decreasing, and the society becomes more aware of climate change. Researchers are investigating these renewable energy sources to find possible solutions for the energy transition. The Ocean Grazer concept is one of these research projects for possible solutions. The idea of the Ocean Grazer is from Drs. W.A. Prins. The Ocean Grazer project is carried out by academic staff, master students and bachelor students of the University of Groningen. The concept of the Ocean Grazer is part of the wave energy converters (WECs). The basic principle of the wave energy converters is that a wave comes in and the system transforms the wave energy into another storable energy. Within the Ocean Grazer, the wave energy is transformed into potential energy of a working fluid.

Ideas of wave energy converters are not completely new. For example, the energy crisis that started in the 1970s resulted in a growing interest in this field. However, since the beginning of the new millennium the field of Wave energy converters (WEC) started to gain interest more than it did before[6]. Developing of a WEC can be done in different manners due to the fact that there are multiple different ideas and concepts. All the different concepts and ideas can be categorized into several categories. The basic categorization of WECs is based on three types of devices. The first category is the terminator category, which consists out of devices with large horizontal extensions parallel to the direction of the wave direction. The second category is the attenuators, which are devices with the same horizontal extensions. However, these horizontal extensions of attenuators are orthogonal to the wave direction, instead of parallel. The third category consists of the point absorber devices. These devices are, in comparison with the other two types, rather small. These point absorber devices can float or can be submerged. Because the devices are rather small, the direction of the wave does not matter.

Besides categorizing based on the type of device, one can also categorize WECs based on their location. The WECs can be placed onshore, near shore and offshore. Onshore WECs are connected to land and are always terminator devices. Near shore devices are placed at depths where the waves will be influenced by the water depth. The devices are often mounted to the ocean floor. The offshore devices are often floating devices and the waves are not influenced by the seabed[6].

Wave energy converters are researched by different institutes. Although the research already began in after the energy crisis in 1970, the technology is not yet mature. A couple of full scale demonstration projects exists, but most of the demonstration projects are still in their research and development phase. Besides, the costs of the produced energy are still too far above their target price.[6] In terms of technology readiness levels, these concepts are at TRL 7. TRL levels are widely used in different sectors. For the Ocean Grazer project, one can consider the TRL levels as defined by the U.S. Department of Energy. The technology readiness levels that the U.S. Department of Energy defined can be found in figure 3[7].

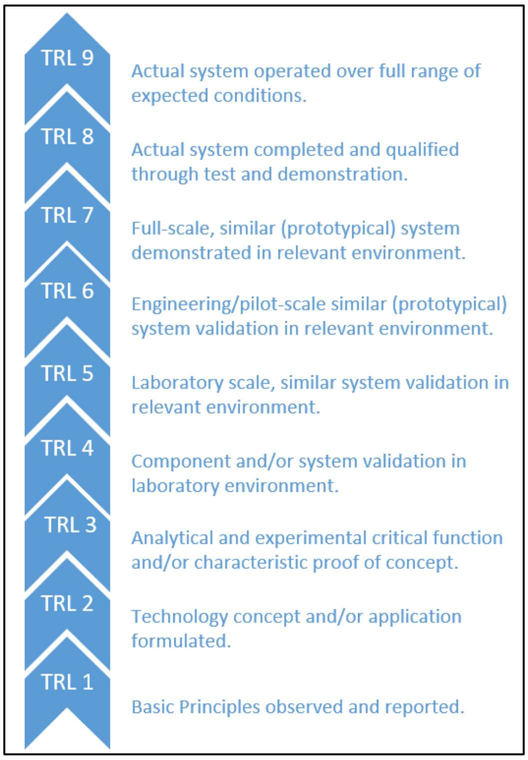


Figure 3: Technology readiness levels[7]

The concept of the Ocean Grazer was patented in 2014. Currently, the project team is working on the third concept of the Ocean Grazer. The first concept was designed to float in the sea. The concept consisted out of two reservoirs, a field of floaters and a platform. The structure had a diameter of approximately 435 meters and a height of 255 meters. The first concept would have been capable of fulfilling the electricity demand of approximately 70.000 households[8].

However, the first concept was too large and too heavy to be able to generate energy. The structure would not have been able to follow the motion of the wave, which is the core of the whole principle. The current concept of the ocean grazer is not a floating structure anymore. The concept is integrated with a windmill base. The pumping system is inside the base of the windmill and the floaters are around the windmill. The pumping system itself, which is the core of this research, has not changed. The current concept is shown in figure 4[9]. In terms of TR levels, one can locate the Ocean Grazer in TRL 4. The project team has two experimental setups. One setup is a setup for the floaters of the Ocean Grazer and one setup for the pumping system of the Ocean Grazer.

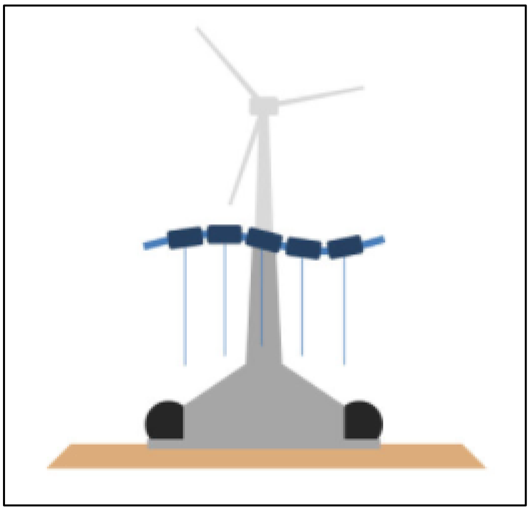


Figure 4: Ocean Grazer concept 3[9]

2.2.2 The working principle of the Ocean Grazer

The system for this project is the multi-piston pump system (MPP), which is a subsystem of the Ocean Grazer. The pumping system is one of the important systems in the energy transformation of the total Ocean Grazer system. The multi-piston pump system is responsible for the energy transformation from wave energy to giving the working fluid a pressure energy. The working fluid is stored, with its potential energy. When there is a demand for energy, the fluid can be allowed to flow back through a turbine. The backflow through the turbine creates electricity, which can be supplied to the power grid.

In the previous section, different elements of the Ocean Grazer were already mentioned. In section 2.2.1 the pumping system is mentioned. The pumping system which is used in the Ocean Grazer is called the multi-piston pump system. The multi-piston pump system is the core system of this research. However, to be able to find the boundaries of the system, it is useful to investigate how the total system (the complete Ocean Grazer) works. In figure 5 the schematic input-output model of the Ocean Grazer is shown.

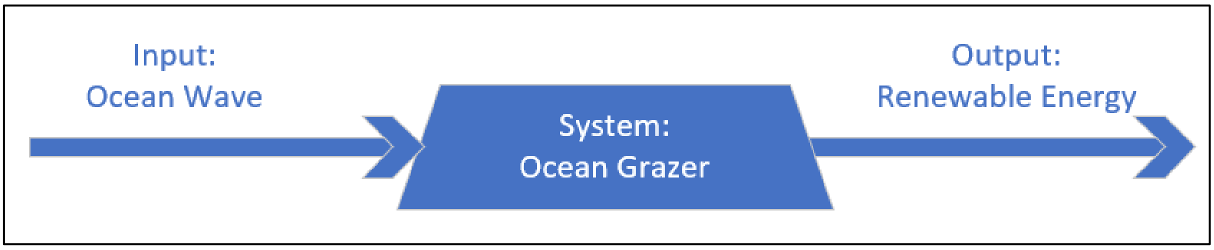


Figure 5: Input-output model of the Ocean Grazer

The Ocean Grazer process starts with an incoming oceanic wave. This incoming wave will move through a field of floaters. The heave of the wave will be followed by the floaters, which will give the floaters an up and down movement. The floaters will transfer their movement to the pistons in the pumping system, to which the floaters are connected by cables. The connection gives the pistons the same up and down movement as the floaters. This up and down movement are called the upstroke and downstroke.

As explained in the previous section (2.2.1), the current concept is placed inside a windmill structure. In this concept, the pistons are pumping working fluid into a reservoir which is outside the structure. This reservoir is called the bladder. Because the bladder is outside the structure, the pressure of the water column above the bladder would push the working fluid back. However, since one can control the backflow of the working fluid, the fluid can be stored in the bladder. The pressure of the water column pushes the fluid back and if the backflow is stopped, the fluid will obtain an amount of energy. When the fluid inside the bladder has this amount of energy, the bladder can be seen as a battery. When one enables the backflow, through a turbine, one can generate electricity which then can be served to the power grid.

2.3 System Description

This section describes the system which is defined for the research. As already mentioned in the previous chapter, the research is about the multi-piston pump of the Ocean Grazer. The previous chapter described the general working of the Ocean Grazer itself, whereas this chapter deals with the working of the multi-piston pump subsystem.

2.3.1 The multi-piston pump (MPP)

As already mentioned in section 2.2.2, the pumping system is placed on the inside of a windmill structure. The first boundary of the system which can be defined is the cable that connects the piston and the floater. Since the research is about the pumping system, this boundary is chosen. The input of the system can be defined as the (up and down) movement of the cable. The second boundary is when the working fluid is stored. The stored working fluid is the end of the process of the pumping system and can therefore be seen as the output of the system. By defining these system boundaries, one obtains the input-output model as seen in figure 6.

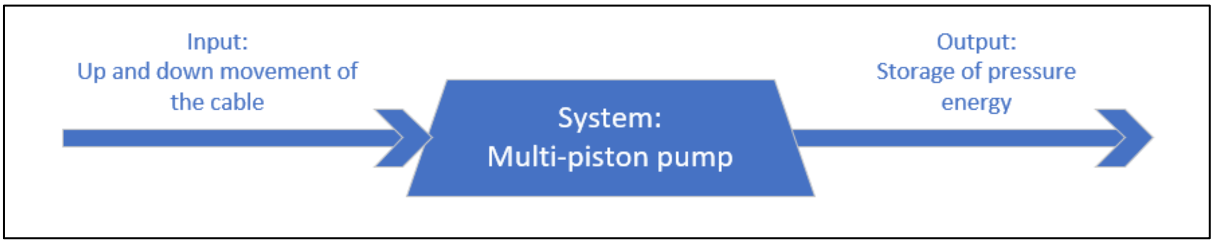


Figure 6: Input-output model of the multi-piston pump

The pumping system consists of several elements. These elements are: check valves, pistons, piston valves, pipes, cables, a control mechanism and working fluid. A schematic representation can be seen in figure 7[10]. The control mechanism is coupled to three pistons and by adjusting the settings of the mechanism, one can control which piston or pistons are used with specific wave characteristics.

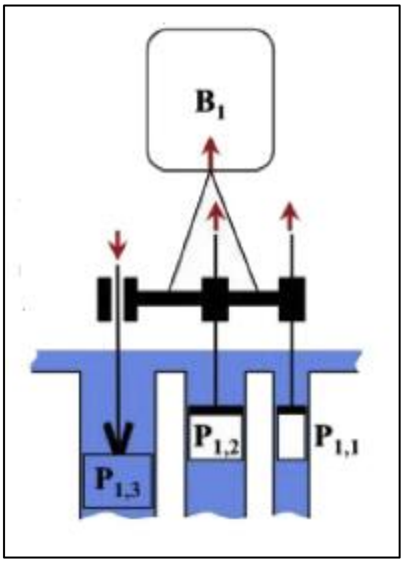


Figure 7: The MPP system[10]

When the cable starts moving up and down, the piston will follow the same motion. As already mentioned, the movement can be divided into the upstroke and the downstroke. During the upstroke, the piston moves upwards. To be able to move the water column above the piston from one position to another, the piston valve must be closed. On the other hand, when the piston moves upwards, it creates a vacuum below the piston. This vacuum opens the check valve, which enables the inflow of new working fluid.

During the downstroke, the process is the opposite of the upstroke process. After the upstroke, the piston is at its highest point. Due to the previous explained vacuum, there is a water column present below the piston. Due to the pressure, which is given to the water column by the down movement of the piston, the check valve closes. Consequently, the piston valve is opened by the water column pressure. This enables the piston to move down to its lowest position, after which a new upstroke can take place.

Before the piston changes from an up movement to a down movement, the motion is stopped for a short amount of time. However, the motion can be stopped quite abruptly. This abrupt stop of the up and downstroke can lead to oscillations in the system. This phenomenon is called slamming and was already investigated in the article of Y.Wei et al[11]. The article states the following: "*High frequency oscillations of the pumping force are observed in the experiment after the switching because the sudden change of the pumping force results in slamming.*" The slamming of the system is currently still being investigated.

To smoothen the transition from upstroke to downstroke and vice versa, the check valves and piston valves were given a special design. The check valve consists out several balls. Below each ball is a hole, which enables the inflow of the working fluid. However, not every ball is the same. There are different sets of balls with different densities. The reason behind this design is that when the pressure increases, the different sets of balls close the holes at different times. In figure 8 both situations are drawn schematically. Figure 8.A. shows the upstroke, whereas figure 8.B. shows the downstroke.

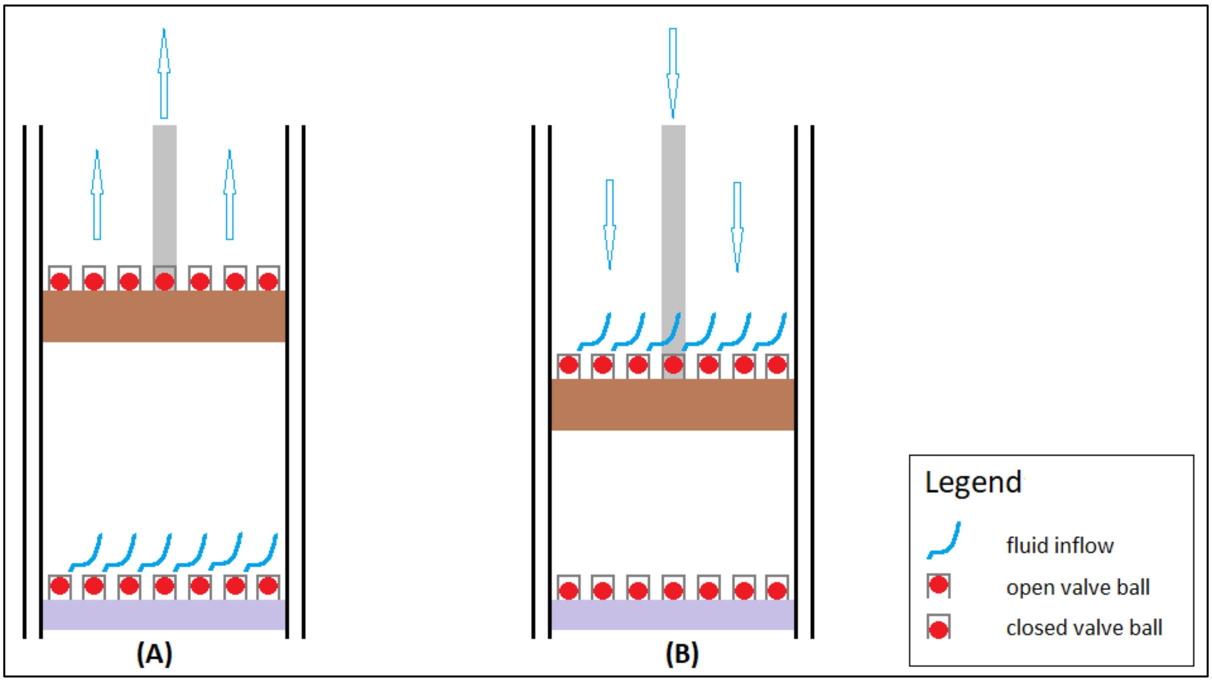


Figure 8.A: The upstroke of the system **Figure 8.B**: The downstroke of the system

2.4 Problem statement

The multi-piston pump is the main subsystem of the Ocean Grazer which is responsible for the transition from wave energy to pressure energy. As explained in the system description, the pump system gives the working fluid a pressurized energy which is stored in the bladder. This bladder is can be seen as a kind of battery.

Currently, only a simplified model of the pumping system exists. This model is based on a single-piston pump (SPP), instead of a multi-piston pump. To be able to model a multi-piston pump system, the team currently uses the cumulative area for the pistons of the multi-piston system within the single-piston pump model. By simplifying the system in this way, one does not take certain specifics of the multi-piston system into account. For example, the end of an up- or downstroke contains an oscillation in the pumping force. In the single-piston model, this is only one oscillation. However, when one has three pistons, and therefore also three oscillations, the oscillations of the three pistons can reinforce each other. The consequence is that the combined oscillation is much stronger than the modelled oscillation of the single-piston model. However, this is not always the case. The oscillations that are produced by the pistons may cancel out.

The problem that the project team encounters with the model of single-piston pump system, is that the model does not take the dynamics of the multi-piston pump system into account.

Important factors in these dynamics are the pumping force and the frequency. Therefore, when modelling the multi-piston pump system with this model is not a fair representation of reality. Therefore, one can write the following problem statement:

"The project team is not able to model the frequency and force behaviour of a multi-piston pump, of both the up- and downstroke, by using the current model."

The problem of not having a model to investigate the behaviour of a multi piston can be seen as a functional problem. A problem is a functional problem, when the problem itself can be directly linked to the output of the system. In this case, the simulation of a multi-piston pump can be directly linked to the output of the system. The output of the system, as described in the system description, is the storage of the pressure energy. This output can be optimized by using the new model.

2.5 Problem owner analysis

The problem statement that was stated in the previous section, has several problem owners. The problem owners that are connected to this problem are all involved in the development of the Ocean Grazer. The first problem owner is M. van Rooij, MSc. Van Rooij is currently the project leader of the Ocean Grazer project. Besides M. van Rooij, Dr. A.I. Vakis is also a problem owner of the problem that was stated earlier. Dr. A.I. Vakis belongs to the academic staff which is working on the Ocean grazer project.

First of all, Dr A.I. Vakis is in charge of the mathematical modelling of the Ocean Grazer system and subsystems. Therefore, he is interested in the outcome of this project, to see if the multi-piston pump can be mathematically modelled. The goal of Dr. Vakis is to be able to model the multi-piston pump system, which is in line with the research.

M. van Rooij, MSc is interested in the optimal settings for the multi-piston pump, with which he is able to let the total Ocean Grazer system capture energy in the most optimal way. Being project leader, he has to sell the idea to possible investors. Therefore, he wants the Ocean Grazer to work as optimal as possible. The model that will be derived in this research can be, together with the existing models, a tool to simulate the working of the Ocean Grazer. To convince investors that the Ocean Grazer works, he has the option to use these models to show that it works.

2.6 Stakeholder analysis

Within this problem, one can define several stakeholders. A stakeholder is a person or organisation that is being influenced (positive or negative) or influences a specific organisation, a government decision, a new product or a project. To determine the stakeholders for this problem one should investigate the questions "Who has a stake in the system?" and "What is the stake they have?".

First of all, Dr. A.I. Vakis is a stakeholder in this problem. As mentioned in the problem owner analysis, he is in charge of the mathematical modelling of the Ocean Grazer project. Since this project should lead to a model that can predict the behaviour of a multi-piston pump, A. Vakis would be interested in the outcome of the project, which gives him a stake in the project. His stake would be that, when having a working model for the multi-piston pump, he would be able to make know how the system would behave.

Second, M. van Rooij is a stakeholder in this project. As also mentioned in the problem owner analysis, he is the project leader of the Ocean Grazer project. The stake that he would have in the project is that the derived model is a step closer to the Ocean Grazer to be manufactured. With the predictions of the system, he will be able to make deals with new partners or investors. At last, Y. Wei is considered a stakeholder for this project. Since R. Zaharia left the project group after his Master thesis, Y.Wei is now in charge of the models which were created by Zaharia. Therefore, he will have a stake in the project as well. Y. Wei his goal is similar to the goal of Dr. A.I. Vakis.

The stakeholders can all have different stakes. The relation of these stakeholders relative to each other can be classified into three types: unitary, pluralist and coercive. The stakeholders that have a unitary relationship have similar values, beliefs and interests. For the stakeholders that have a pluralist relationship will have no similar values and beliefs. However, pluralist stakeholders do share their basic interests. The stakeholders that are defined as coercive have few interests in common and would hold conflicting values and beliefs.[12] In the case of the multi-piston pump project the stakeholders can be seen as unitary. All four stakeholders have the same interest, which would be a working model for the multi-piston pump. Besides, they share similar values and beliefs about the project. Table 1 shows the stakeholders their goal, their stake and the possibility of conflict.

Table 1: Stakeholders

Stakeholder	Stake	Goal	Conflict (yes/no)
Dr. A.I. Vakis	Has a stake as being responsible for the mathematical modelling	Having a working model to simulate the MPP	No
M. van Rooij, MSc	Has a stake as project leader	Having a tool to predict the working of the MPP	No
Y. Wei	Has a stake as being responsible for the models after R. Zaharia left	Having a working model to simulate the MPP	No

2.7 Goal and research questions

2.7.1 Goal of the research

Within the research, a model is created to analyse the behaviour of the multi-piston pump under different wave circumstances. This model is an adjustment of the model that was already composed and will show what happens to structure of the system under influence of the different wave circumstances.

Therefore, the goal of the research can be derived as follows:

"The goal of the research is to design a model of the multi-piston pump to analyse the frequency and force behaviour of the pump."

2.7.2 Research questions

Looking at the problem definition and the goal of the research, one can derive the research question for this project. This research question is the following:

"How does a multi-piston pump influence the frequency and force in the pump system for both the upstroke and downstroke?

To be able to answer this question, several sub research questions can be derived.

1. How can the model be adjusted from an SPP model to an MPP model?
2. How can the force output be interpreted?
3. In what way do the old and the new model differ from each other?

3. Research Design

3.1 Methodology

3.1.1 OBS method

During this research the OBS method is used. The OBS method is an alternation of the regulative cycle. This method is described and explained in the course *Business Systems Design* (BSD) given by Drs. W.A. Prins at the University of Groningen. The important key element of this particular method is that it sees problem solving as (re)designing systems to improve the functioning, as said during lecture 1 of BSD. The OBS method consist of multiple phases. These phases are the following:

- Diagnose
- Design
- Implementation
- Evaluation

Each of these phases is an iterative cycle on itself. Which means that throughout the each of the phases, all the steps are repeated. Besides the different phases, the whole OBS method is iterated. For example, it could be that within the design phase something does not add up and one must check and maybe alter the previous steps. The diagnose phase consists out of three different steps. These steps are the Problem/Goal determining step, the Conceptual research design and the Empirical diagnose.

Within the OBS method, one chooses one or multiple approaches for the system. To make this choice, the type of relation between the stakeholders is taken into account as well as the system being simple or complex. In figure 9 the matrix is shown which provides the different approaches[12].

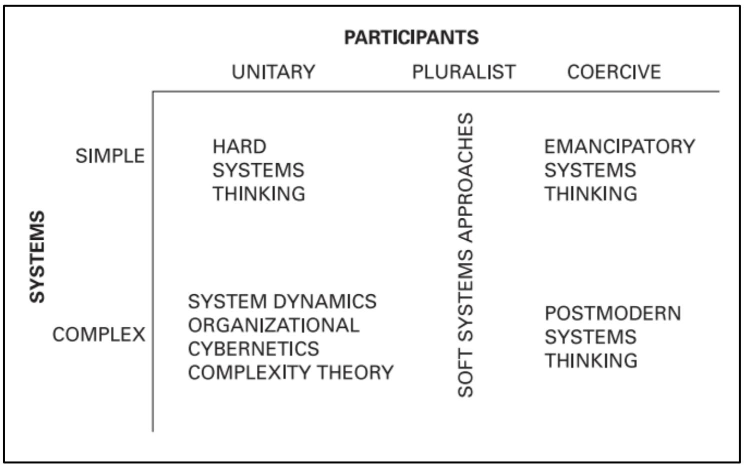


Figure 9: The system approach matrix[12]

As already mentioned in the stakeholder analysis, the relationship between the different stakeholders can be classified as unitary. The system is classified in a simple system or a complex system. The distinction between a simple and a complex system is based on the number of subsystems which are involved. A simple system has a few subsystems which have highly structured interactions. On the other hand, a complex system has many subsystems which have loosely structured interactions. The system of the multi-piston pump can be seen as a simple system. Therefore, the system approach that would be the most applicable to this project is the *Hard Systems Thinking* approach.

As stated before, the OBS method is used throughout the research. However, within the design of the new model, one can use the design cycle. The design cycle, also called engineering cycle, is widely used for technological projects and consists out of five steps[13]. This cycle is shown in figure 10.

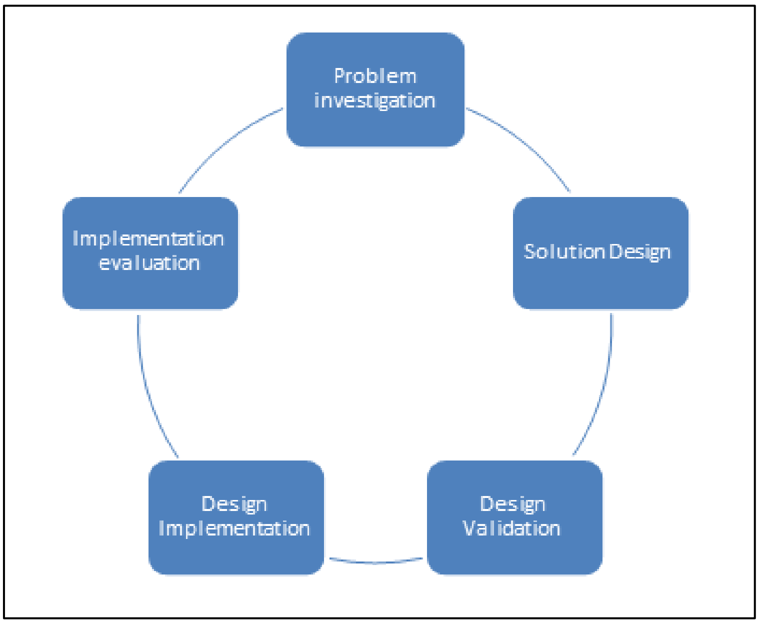


Figure 10: The design cycle by R. Wieringa[13]

The design cycle, which will be used to design the new model, can be placed within the OBS method. The cycle will cover the design phase of this method. The emphasis of the design phase of the OBS method will, therefore, be on the use of the design/engineering cycle as explained before.

3.1.2 Research Design Steps

As already mentioned in the method section, the approach that would be applicable for this system is the *Hard Systems Thinking*. The hard systems thinking approach consists out of multiple steps itself. This is shown in figure 11[12].

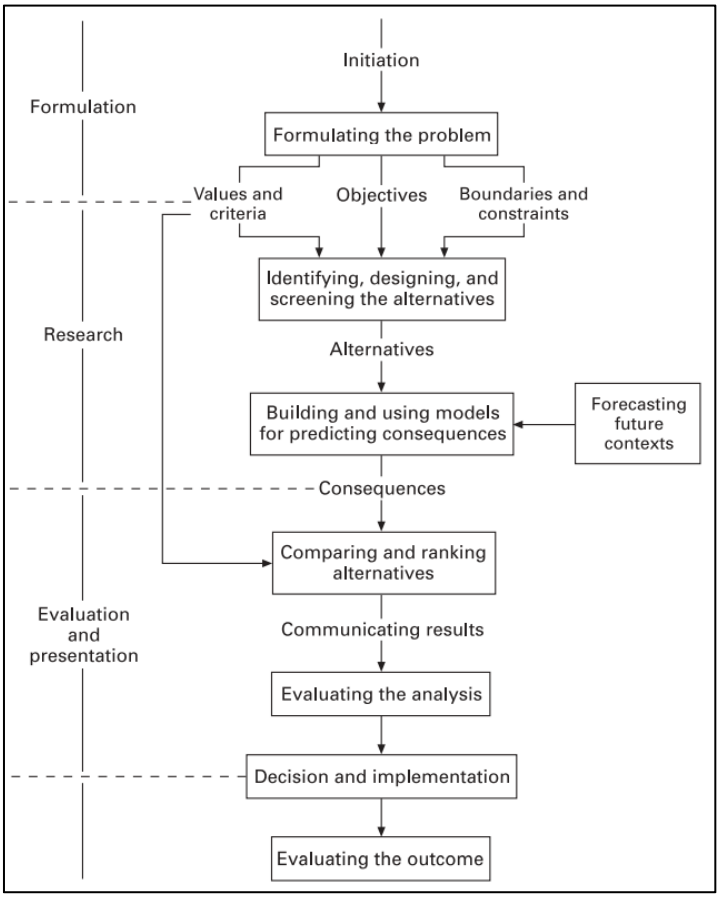


Figure 11: Hard Systems Thinking approach[12]

The steps from the approach can be incorporated into the research design steps. Besides the fact that they can be incorporated, they can also be iterated throughout the research. With the contents of this approach and the sub research questions, one can setup the research design steps.

Step 1: Understanding the Ocean Grazer concept and the problem.
Step 2: Identifying the problem environment and key players.
Step 3: Defining the problem, the goal and the research question(s).

Step 4: Understanding the currently used model.
Step 5: Gathering relevant literature.
Step 6: Deriving conceptual model.

Step 6: Redesigning the available model.
Step 7: Evaluating the model output.
Step 8: Evaluate the redesign.

The research design steps are in line with the four sub research questions that were formulated in chapter 2.7.2. The first sub research question can be related to step 4 in the research design steps. Step 6 and 7 are related to the second and third sub research questions. The last sub research question is related to the last step in the research design steps, which is step 8.

3.2 Resources

3.2.1 Data acquisition

To be able to start the project, several different data types are needed. Each type of data is gathered in a different manner. The data types that are needed are: parameter data, parameters of the other two pistons and literature data. The literature data acquisition is described in the next section.

The parameter data of the experimental setup can be found in the *Matlab* script written by R. Zaharia[14]. The parameters that will be extracted from his research, are the parameters that have not changed in the experimental setup. If the parameters are changed or if parameters are added to the system, the new values will be acquired from the experimental setup and extracting them from the Solidworks model. The same holds for the parameters of the 2 pistons that have to be added to the model.

3.2.2 Tools

The project for the multi-piston pump involves experimental and numerical models to analyse the behaviour of the system. The type of models that are used are numerical and mathematical models. To be able to perform the research, several tools were used for these models. The tools that were used provided a platform to model and develop a mathematical model for the multi-piston pump.

The conceptual model is the first tool that is used in this research. The tool is used to obtain a higher understanding of the system. There are different types of conceptual models. In this research the entity-relationship modelling is used. This type of conceptual modelling shows the important components of the system and their relationships. By investigating the different relations of the systems components, one can see more easily where the system can be adjusted to influence the outcome.

The second tool that is used is the software of *Matlab*. *Matlab* is used for the script which is coupled to the currently used SPP model. Within the script, that was made by R. Zaharia, the different parameters of the system can be defined. Besides the parameters, the script gives also an option to give a set of graphs as an output. The exact working of the script and the model will be given in the first section of the literature research (4.1).

Another tool that is used is the *Simulink* software. Both *Simulink* software and *Matlab* software is made by the company *MathWorks*. The *Simulink* software is used to create the model of the SPP and will be used to create the MPP model. As said, the currently used model is created in *Simulink* and therefore the MPP model will also be built in this software.

The last tool that will be used in this research is the Solidworks software. The project team has built a model of the experimental setup in this software tool. Therefore, when parameters are missing, they can be extracted from this model. Besides the parameters, the software can also be used to validate calculations.

3.3 Conceptual model

The conceptual model is a tool to obtain a better understanding of the system. It shows different factors that are influencing the system and their relationships to each other. For the conceptual model that was made for this research has a few assumptions.

1. The pistons have the same effect on the next block in this schematic, so they will be considered as one block.
2. The lower check valves have the same effect on the next block in this schematic; therefore, they will be considered as one block.
3. Assumption 2 holds also for the piston valves.

The conceptual model of the MPP system is shown in figure 12.

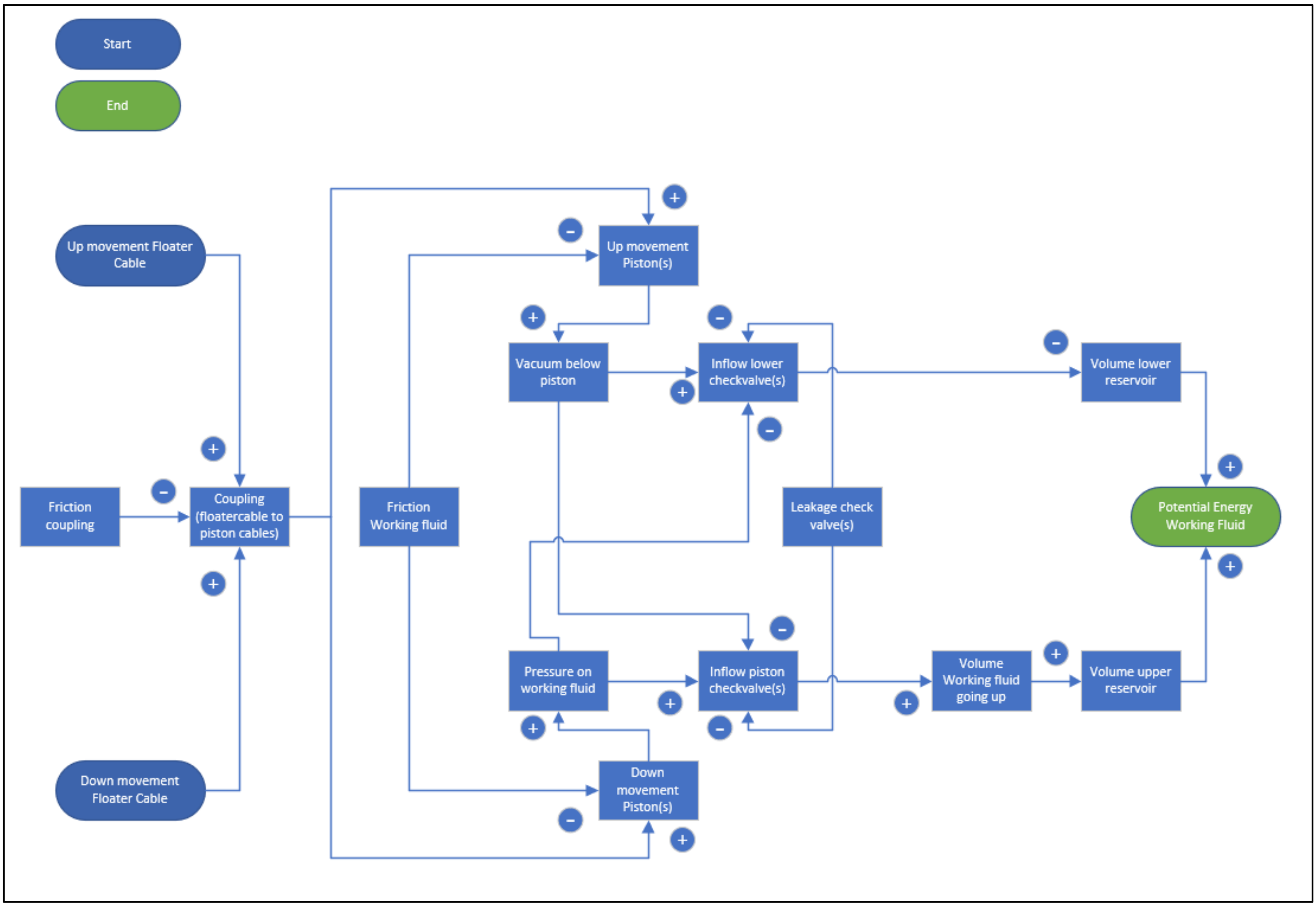


Figure 12: The conceptual model of the multi-piston pumping system

As mentioned before, the conceptual model shows the relations between different components of the system. These relations between components is important when modelling the system into a Simulink model. Therefore, when building the model for the MPP in Simulink, one knows how the different elements are affecting each other, which makes the modelling of the system less difficult. In the conceptual model, the components are shown by blocks, the connection between components with arrows and the relation between the components are denoted with a plus sign or minus sign. The plus sign denotes a positive relation, whereas the minus sign denotes a negative relation. Furthermore, the start of the system is denoted by a rounded blue block and the end of the system by a green rounded block.

3.4 Risk analysis

Before the research was started, a risk assessment had to be performed. Within the risk assessment the risks of the project are explained and analysed. There are different risks for the different stakeholders, but also for the researcher.

The first risk that was considered was that the developing of the model takes too much time. If the developing takes too long, the whole project could not be finished on time. This would mean that the project had to be stopped, and the researcher would have to finish his project later or had to start a new project.

The second risk that was considered was that the developed model would not work. In this case the Ocean Grazer Project team would not have been able to model and analyse the behaviour of the multi-piston pump. The project leader and project manager would therefore not have a tool to predict the behaviour of the Ocean Grazer. Consequently, they might not have been able to get potential investors on board with their company. Besides the project team, the researchers might not have had enough time to solve the problem.

Another risk that could be assigned to the project is the risk that the researcher was not being able or allowed to finish the research. This could have been due to a rejection at the GO/NO GO moment in earlier stages of the research. The risk for the Ocean Grazer group was that there still would not be a working model for the multi-piston pump. The consequence would be that the whole Ocean Grazer project would have been delayed.

The last risk that could have interfered with the project is that the results of the research would fall short to the expected results. In this case the project team of the Ocean Grazer has to invest time and money in new research. The consequence for the project team is that the whole project would be delayed or that the whole system has to be redesigned.

4. Literature study

4.1 The current model

The currently used model is developed by R.M. Zaharia, who was part of the Ocean Grazer project team. In this Master thesis a description was given about how the model of the single-piston pump was build and how it works. To be able to adjust this model for the multi-piston pump, one must understand how the current model was derived and how it works[14].

4.1.1 The experimental setup

The model, which is currently used for simulations, is based on the experimental setup, which is currently situated in the water hall at the Rijksuniversiteit Groningen (figure 13). The water hall is part of the Nijenborgh building at the Zernike Campus. The setup consists of several elements, namely:

- Pipes
- The upper reservoir
- The lower reservoir
- The check valves
- Pistons
- Piston cable
- Rotary motor

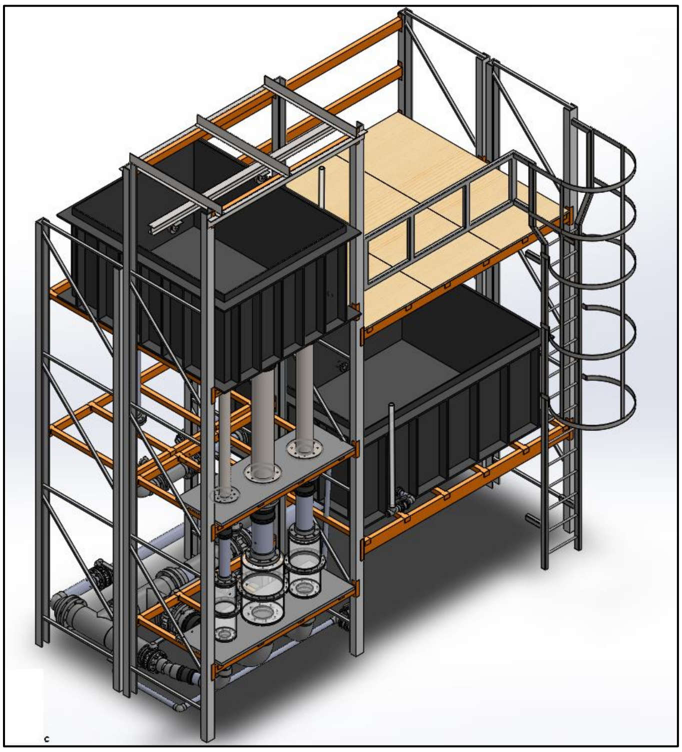


Figure 13: The experimental setup

The two reservoirs are connected by several pipelines, the check valves and the pistons. To be able to move the fluid from the lower reservoir to the upper reservoir, one needs an input. In the concept of the Ocean Grazer, the input is an oceanic wave. However, in the case of the experimental setup the input is a rotary motor which simulates the wave motion. The drawback of using this rotary motor, is that it acts as an ideal wave. Currently the experimental setup can be only used for simulating a single-piston pump system.

The experimental setup works according to the same principles as the Ocean Grazer concept, except for, as already mentioned, the wave input. In this case, the motor is the driver of the system which is connected to the piston cable by means of a steel rod. To be able to simulate different type of waves, one can adjust the frequency of the motor and one can adjust the length of the rod. Adjusting the length of the rod can be done by using the holes in the rod, which each represents a specific amplitude. By using different combinations of motor frequency and the length of the rod, one can simulate many different types of waves.

The piston of the system is connected to the motor by means of a cable and the "amplitude-rod". However, to be able to measure the force that is exerted while pumping, a force sensor is placed between the cable and the rod. As previously described in the system description, there are two movements: the upstroke and the downstroke. During the upstroke the piston moves upwards, the piston valve is closed, and the check valve is open. During the downstroke the opposite occurs: the piston moves downwards, the piston valve is open, and the check valve is closed. With each upstroke, working fluid is pumped from the lower reservoir to the upper reservoir.

The check valves in the experimental setup consists of multiple balls. As explained in the system description, these sets of balls have different densities. The reason behind this design choice is the slamming phenomenon which occurs at the end of each up and downstroke. The slamming is caused by the sudden stop of movement of the piston. Due to the sudden stop of motion, high frequency oscillations can be seen in the pumping force.[11] The check valve of the experimental setup consists of 12 balls, whereas the piston valve consists of 4 balls. The difference in density of the different set of check valve balls will enable the system to have a smoother transition from the upstroke to the downstroke.

4.1.2 The model

Overview

The model created by R. Zaharia is based on the experimental setup which is described in the previous section. The model itself consists of multiple elements, four Matlab scripts and one Simulink model. The Matlab scripts are written to assign parameters to the different modeling blocks (*initialization_parameters.m*), to let the Simulink model run (*run_simulink-model.m*), to calculate specifics of the system (*calculationsSimulink.m*) and to plot the different outcomes of the model (*plot_simulinkresults.m*).

The model is built in different layers. The reason behind the different layers is to have a clear overview of the model and how it works. The different subsystems are, for example, presented as one block in the layer of the total system. When opening the subsystem, the modelling blocks of this subsystem are shown in the subsystem layer. By modelling the system in this way, the model is less chaotic and more clarifying. The first layer consists of one block and three graphs. These graphs show the change over time of three important parameters; the pumping force, the upper reservoir water level and the lower reservoir water level. The white block is the link to the second layer, which contains the block model of the pumping system. Figure 14[14] shows the first layer of the model.

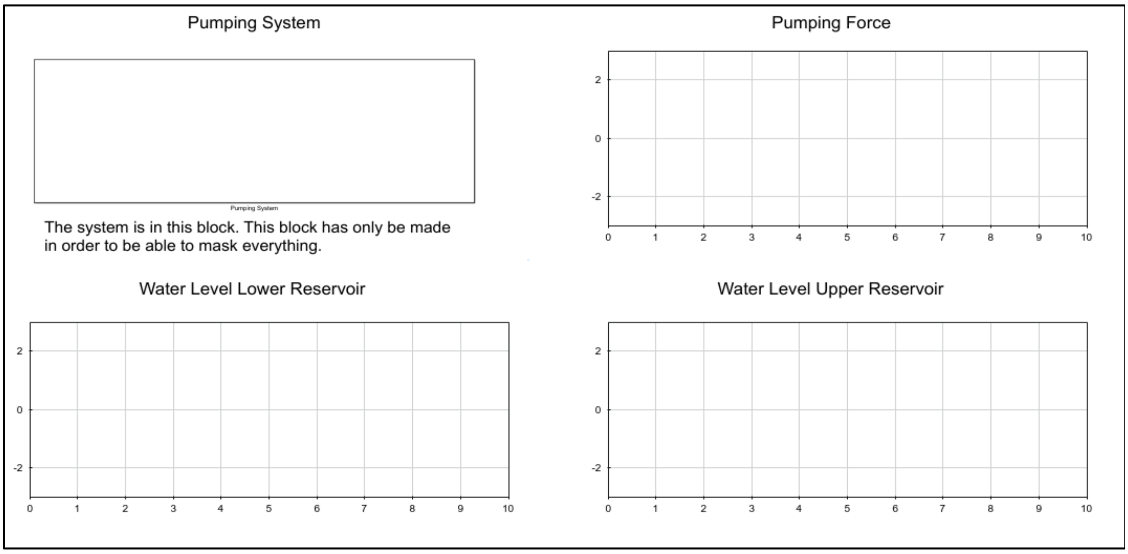


Figure 14: The overview of the model[14]

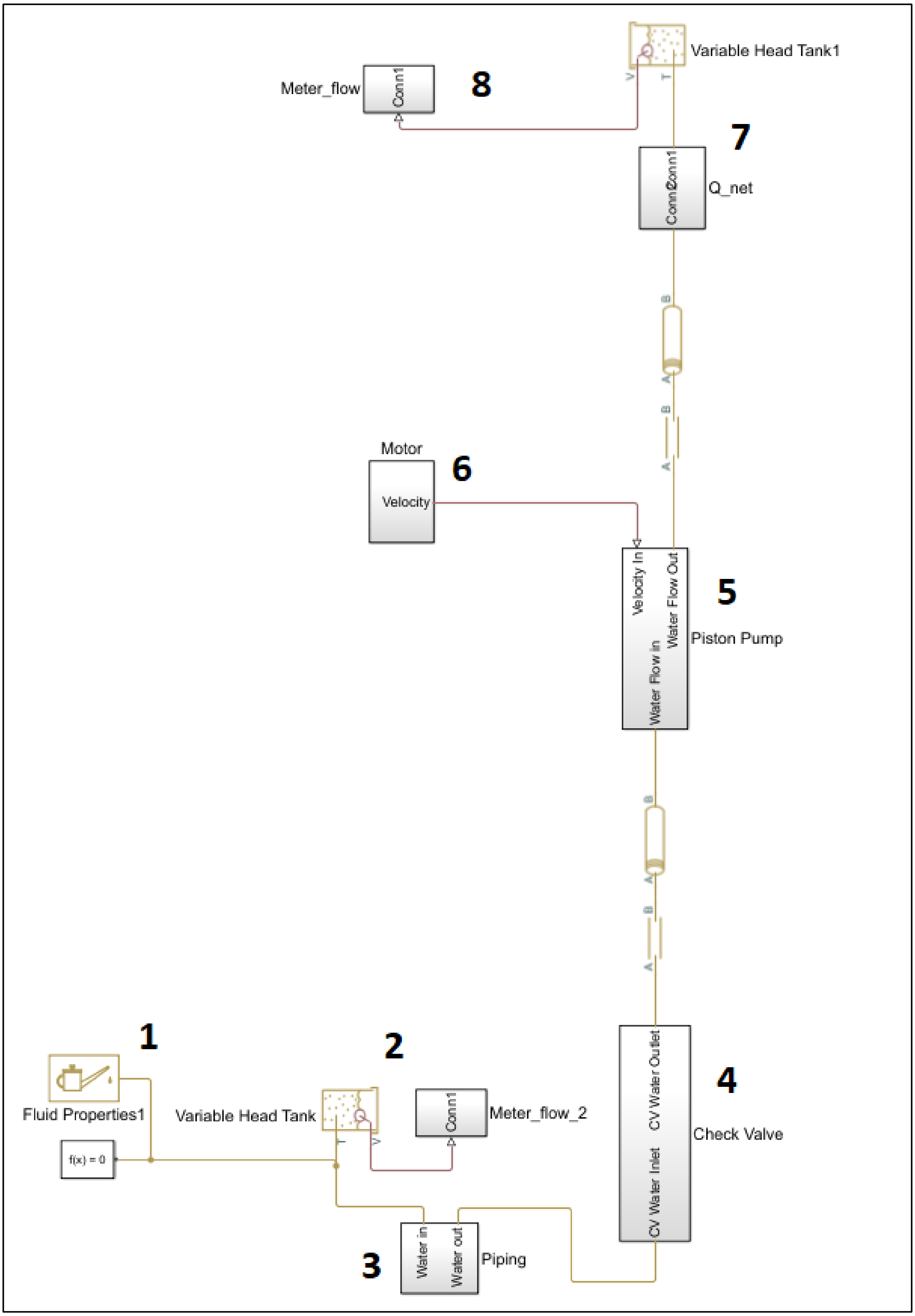


Figure 15: The SPP Simulink block model[14]

Complete SPP model

The pumping system model is shown in figure 15[14] on the previous page. The blocks are divided in sections in the figure, which are numbered from 1 to 8. The numbering of these blocks follows the flow of the working fluid in the system, from the lower to the upper reservoir. The gray blocks represent the sub systems which are modelled on the next layer.

Section one of the model contains two blocks. The first block is called *"Fluid Properties 1"* and the second block is called *"f(x)=0"*. The first block assigns the properties of the working fluid to the model. These properties are for example the temperature of the fluid, the density and the viscosity of the fluid. The second block is a so-called solver configuration block, which defines the settings that are used for the simulation. As stated by the *Help*-section of Simulink, it specifies the solver parameters that a model needs before it can start the simulation.

The second section contains two blocks as well. These blocks are the *"Variable Head Tank"* and the *"Meter_flow_2"*. The variable head tank block models a pressurized fluid container with a variable fluid level. The block will therefore model the lower reservoir of the experimental setup. The meter flow block represents the flow sensor that measures the water level in the lower reservoir.

Pipelines

The third section contains a grey subsystem block. The subsystem that is modelled in this block is the pipeline section from the lower reservoir up to the check valve. Opening the grey subsystem block, the pipeline block model is shown. Figure 16 shows this block model.

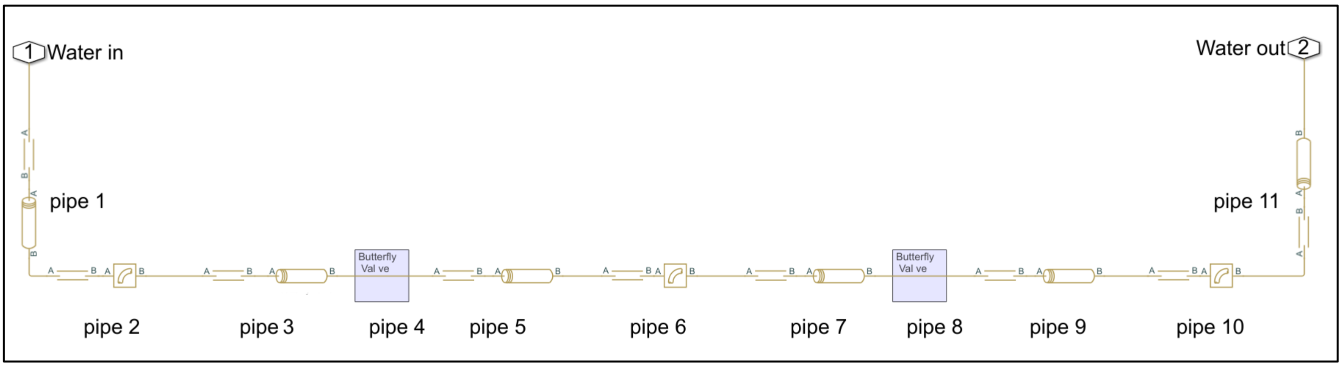


Figure 16: The inflow pipeline subsystem[14]

In the model every pipe has two modelling blocks. The first block of each pipe accounts for the fluid inertia, whereas the second block models the pipe specifications. The reason behind the modelling of the pipe in this way is that the pipe block does not account for the fluid inertia. The fluid inertia models the pressure differential, which is a consequence of the change in fluid velocity. The pipe block models the different parameters of the pipe itself, for example: the diameter, the length, the cross-section type and, in the case of a bend pipe, the bending radius. The parameters of the pipe are defined in the Matlab script *initialization_parameters.m*, which are directly coupled to the Simulink model.

The check valve

Section number four contains the grey subsystem block of the check valve. The check valve is the part which enables the inflow of working fluid to the piston. Opening the subsystem block, the block model of the check valve is shown (Figure 17).

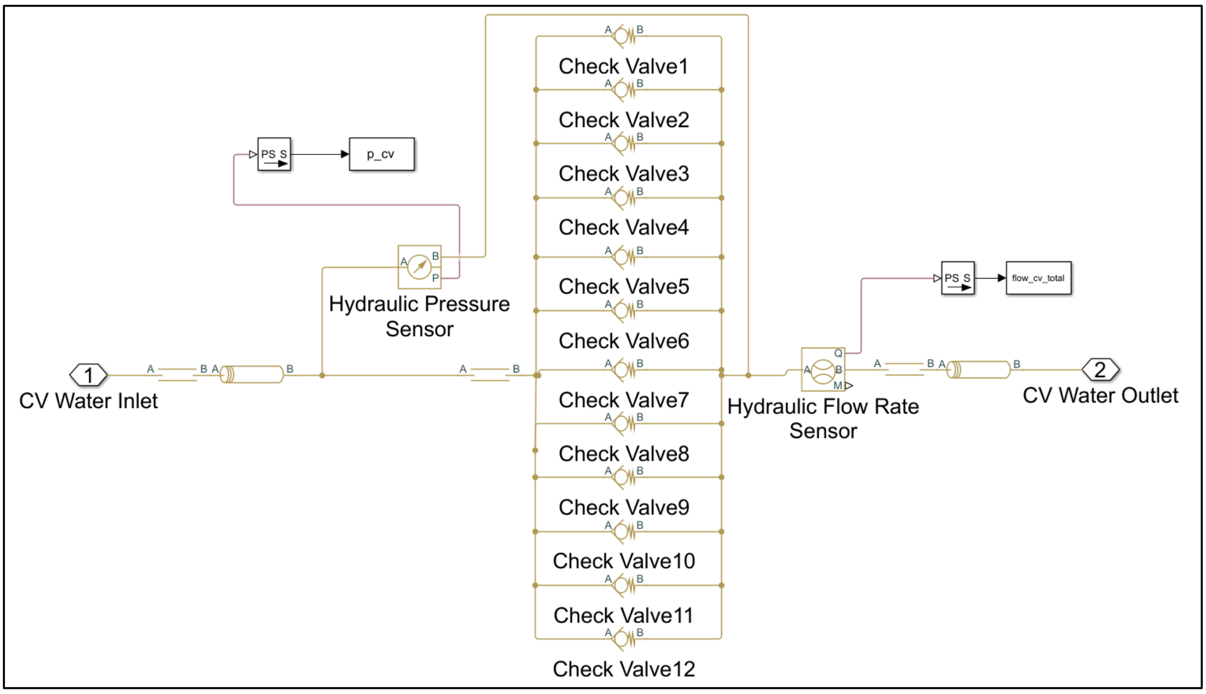


Figure 17: The check valve subsystem[14]

The subsystem model of the check valve starts at the *"CV Water Inlet"*. This block is connecting the signal of the above lying layer with this layer. Since the check valve is placed inside a pipe, the pipe must be modelled as well. This pipe is modelled in the same way as the pipes in the subsystem of the pipeline, first a fluid inertia block and after that the pipe characteristics block. Since the working fluid must pass through the holes of the check valve balls, the passage area becomes smaller. The consequence is that there has to be another fluid inertia block, with the passage area of these holes. The balls of the check valve are each modelled as an individual check valve. The check valve block in Simulink has only one ball, therefore, the model must contain 12 individual check valve blocks. After the check valves, there are again two pipe modelling blocks and the *"CV Water Outlet"*, which connects the signal again to the above lying layer.

The modelled subsystem contains two sensors, a hydraulic pressure sensor and a hydraulic flow rate sensor. The pressure sensor measures the pressure difference over the check valve, whereas the flow rate sensor measures the flow through the check valve. Both measured signals, pressure difference and flow, are connected to a block which converts them from a physical signal to a Simulink signal. The signal from Simulink is on its turn send back to the Matlab workspace.

The piston

Before the signals go through section five, another pipe is modelled. Although the pipe section in which the piston moves is modelled inside the piston block, the connected pipes are modelled in below and above the piston subsystem block. The model of the piston is shown in figure 18.

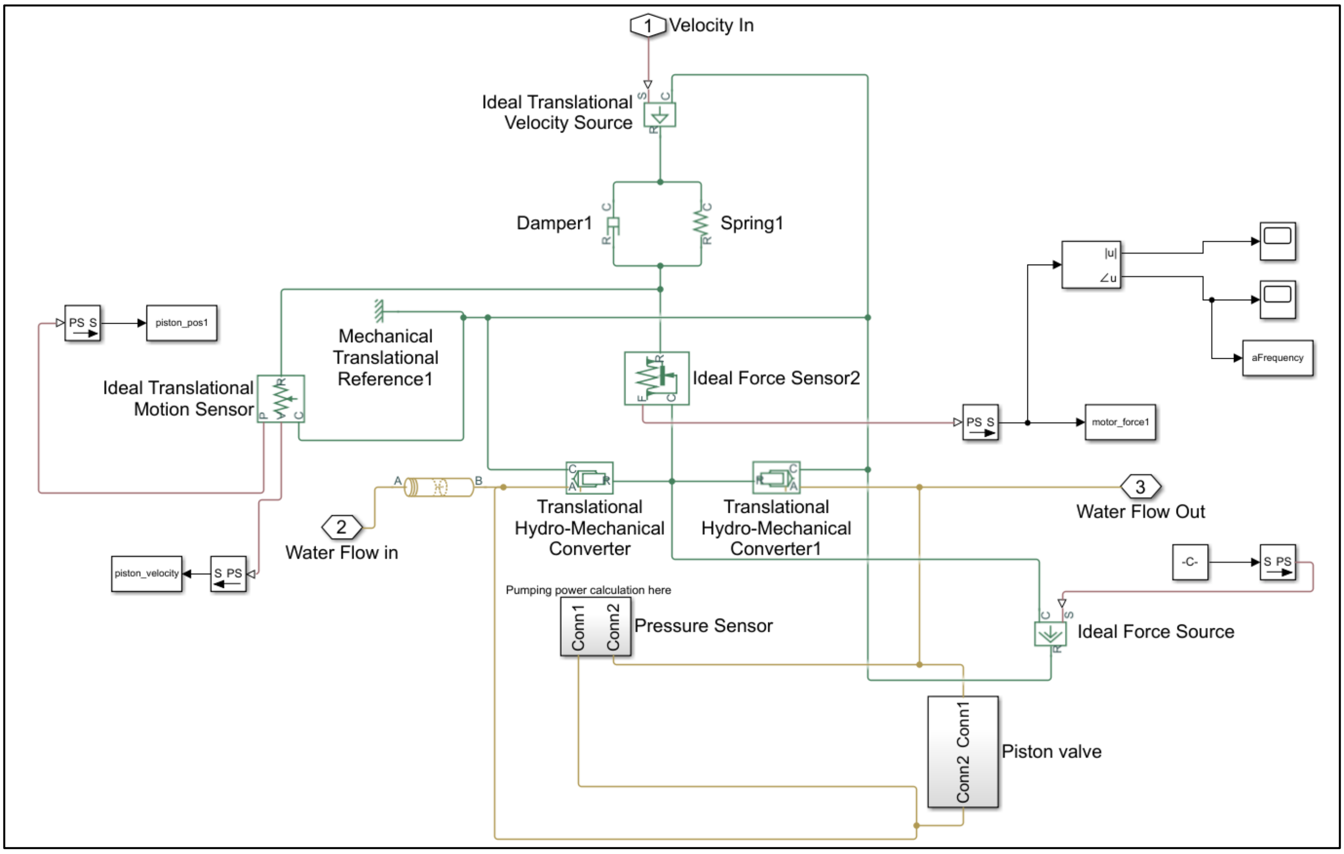


Figure 18: The piston subsystem[14]

The piston subsystem is modelled in three distinct colours, namely green, yellow and red. Each colour represents a different type of system. The green lines represent the mechanical system, the yellow lines represent the hydrodynamic system and the red line represents the signal system. The subsystem of the piston is more difficult than the other blocks, since it uses a large amount of modelling blocks.

There are two physical signals that are connected to the above lying layer. These signals are the velocity and the water inflow. Starting with the mechanical system, the velocity signal is connected to an ideal translational velocity source. This block assures that the system *"maintains the specified velocity regardless of the force exerted on the system"* according to the Simulink Help function. The now ideal velocity is applied to a spring and damper system. This spring and damper system represents the cable which connects the piston to the motor. The reason behind modelling the cable as a spring-damper system is that the cable can be elongated (behave as a spring) and it can damp the motion (behave like a damper). The motion that it makes is connected to two sensors, one *Ideal Force Sensor* and one *Ideal Translational Motion Sensor*. With these sensors, one can measure the force acting on the system and the position and velocity of the system respectively. The signals are, just as the sensor signals in previous subsystems, converted to a Simulink signal and transported to the work-space in Matlab. The ideal force sensor is coupled to a block which transforms a physical signal to a Simulink signal, in this case the motor force. However, by adding a Fourier trans-form block, one can also measure the frequency of the system.

The signal of the Water inflow is connected to a segmented pipe section. This section models the pipe in which the piston moves. The piston itself is modelled by the two *Translational Hydro-Mechanical Converters*. These blocks represent *"an ideal transducer that converts hy-draulic energy into mechanical energy in the form of translational motion of the converter output member."* There are two of these blocks because one block is only able to model the movement of one direction[14]. Both blocks have to be connected to the velocity source, oth-erwise the piston will not move. Inside the segmented pipe the piston valve and the pressure sensor are placed. The piston valve is modelled in the same way as the check valve, the only differences are that the piston valve consists of 6 balls, or independent valves, and the out-put parameters (pressure and flow) are called differently. The ideal force source is connected to a constant (the *-C-* block) which has the value of 9,81*25. This value is the gravitational force of the mass of the piston.

The motor

In section six, the motor is modelled. This subsystem is quite a simple block and is shown in figure 19. The input for this block is a sine wave, which parameters are found in the *initiali-zation_parameters.m* script. The sine wave is a Simulink signal, which is converted to a physical signal. The physical signal is the velocity of the system and is connected to the above lying layer of the model. Besides just converting to a velocity signal, the sine wave is integrated to obtain the position of the motor and it is differentiated to obtain the accelera-tion of the motor. Both parameters are transferred to the workspace of Matlab.

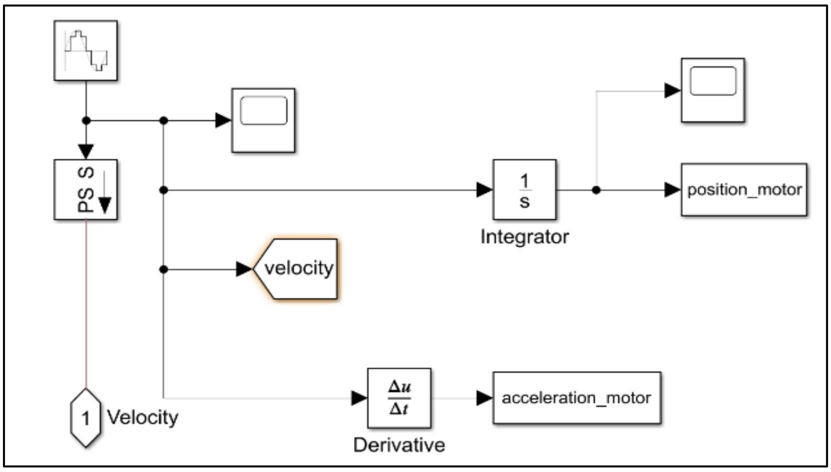


Figure 19: The motor subsystem[14]

The tank and flow meters

Section 7 covers the *Q_net* flow meter. This is a very simple block in comparison with all the other blocks. The subsystem contains only a hydraulic flow rate sensor and a physical signal to Simulink signal converter to let the *net_flow* parameter be transferred to the Matlab work-space. The last section, section eight, consists of the last two remaining blocks. These blocks are second *Variable Head Tank*, representing the upper reservoir, and the *Meter_flow* which is the sensor measuring the water level in the upper reservoir.

4.1.3 Outcomes

The model that was described in the previous paragraphs has several outputs. These outputs vary from piston forces to the pressures and flows through the valves. In the script *plot_simulinkresults.m* the different outputs are plotted against the time of the simulation. In the script, several switches are used to be able to let Matlab plot the figures that are needed. Before the different graphs that are plotted, an initial calculation script calculates different characteristics of the total system. This is shown in figure 20.

```
---------------------------------------------------------
|           Results Simulink Model            |
-------------------------
Average Max displacement = 1 m
-------------------------
Total Pumping energy = 3144.2734 J
Total Potential energy = 956.9508 J
Avg. Pumping energy per stroke = 1048.0911 J
Avg. Potential energy per stroke = 318.9836 J
-------------------------
Volumetric Efficiency = 82.7736 %
Mechanical Efficiency = 36.7686 %
Total Efficiency = 30.4347 %
```

Figure 20: Results of the calculation script

The calculation script shows the different energies that are generated and used by pumping the working fluid. As can be seen, the total pumping energy is almost three times the total potential energy which is stored. Besides the total energy, the average energy per stroke is calculated. Using the energies generated, the different efficiencies are calculated.

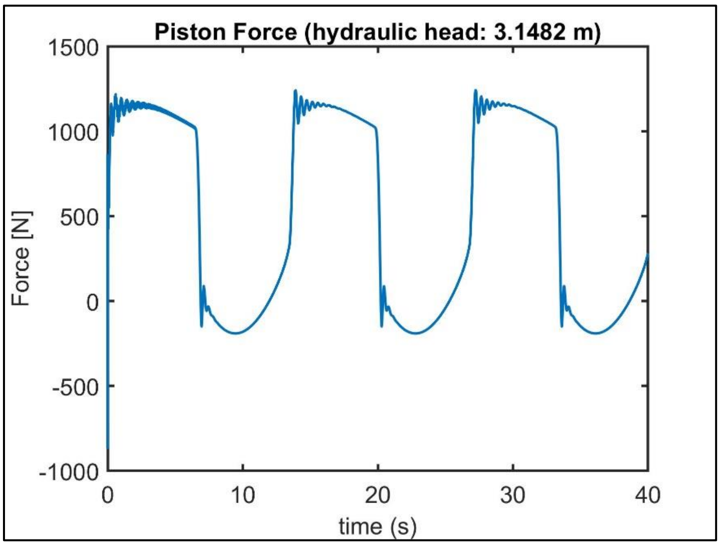


Figure 21: The piston force

The first plot that is shown is the piston force of the system. In figure 21 the piston force is plotted against the time of the simulation. This graph can be used to compare the values for the forces of the piston which is currently used in the experimental setup. The line of this piston should have the same shape in the old and new model. The graph shows the pumping force over the course of three upstrokes and downstrokes. Besides the force and the up-stroke and downstroke, the plot also shows the slamming which occurs at the end of both up and down stroke. The slamming is indicated by the oscillations in the piston force.

The piston movement is made possible by the inflow through the check valve and through the piston valve. The flow through both the check valve and the piston valve can be plotted against the time, this is shown in figure 22. When these two graphs are plotted, one can immediately see the dynamics, in terms of closing, between the check valve and piston valve. In the upstroke, the check valve opens and enables a flow through it. In the downstroke the piston valve is opened and enables a flow through it. In both cases, when one of the valves is open, the other is closed. The calculations script of Matlab already showed that the average maximal displacement equals 1m. This value is supported by the graph of the displacements figure 23, which shows a sinusoidal wave function with maximum values of 1m in terms of the displacement.

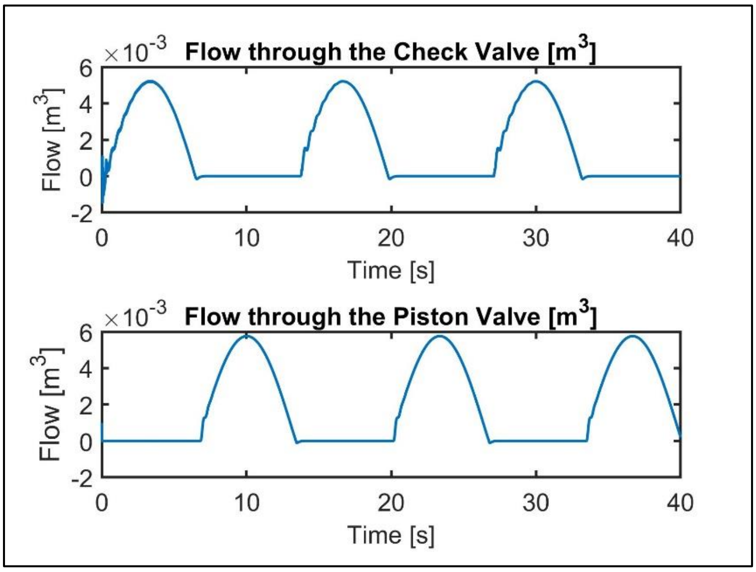


Figure 22: Flow through the check and piston valve

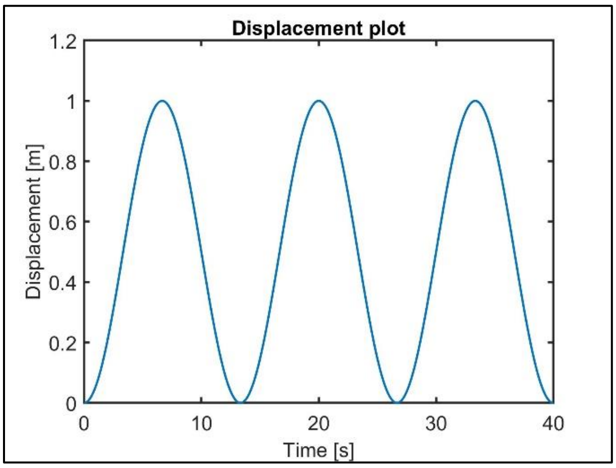


Figure 23: The displacement plot

4.2 Oscillating systems

The system of a multi-piston pump involves several cables. One cable is involved in the connection of the floater and the coupling mechanism, whereas three cables are involved in the connection of the three pistons with the coupling mechanism. These cables are not ideal motion transformers. As the up and down motion of the piston is performed, the cable can elongate but also damp the motion. This behaviour of the cable can therefore be seen as a spring-damper system.

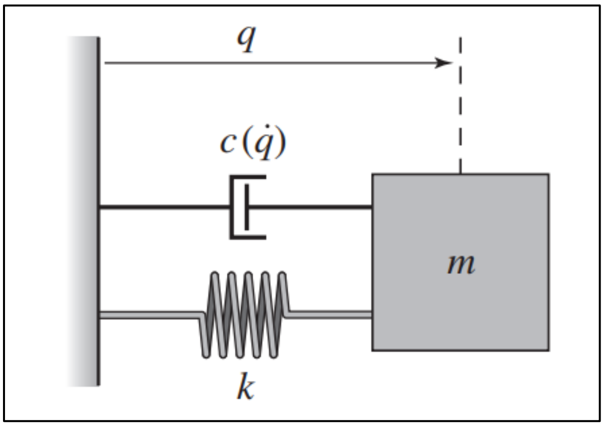


Figure 24: Spring damper system[15]

A spring-damper system can be seen in figure 24[15]. In this figure, the following denotation is used:

1. k: spring constant
2. c($\dot{q}$): damping function
3. m: mass
4. q: displacement
5. $\dot{q}$: velocity

Using Newton's second law ($F = ma$) and the given parameters. One can obtain the following differential equation:

$$F_{external} - c(\dot{q}) - kq = m\ddot{q} \tag{1}$$

The dots above the q are denoting the first or second derivative of the displacements, hence $\dot{q}$ equals the velocity and $\ddot{q}$ equals the acceleration. Besides, the damping function is often described as $c \cdot \dot{q}$. Therefore, one can rewrite the equation to:

$$F_{external} - c\frac{dq}{dt} - kq = m\frac{d^2q}{dt^2} \tag{2}$$

This equation can be rewritten in terms of radian frequency (ω_0) and damping ratio (ζ). These two terms have the following equations:

$$\omega_0 = \sqrt{k/m} \quad \text{and} \quad \zeta = c/(2\sqrt{km}) \tag{3) (4}$$

From the radian frequency one can determine the natural frequency of a system (f_n):

$$f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \tag{5}$$

Combining the equations for radian frequency and the damping ratio with the differential equation, the equation can be rewritten as:

$$\frac{d^2q}{dt^2} + 2\zeta\omega_0\frac{dq}{dt} + \omega_0{}^2q = F_{external} \tag{6}$$

The value of the damping ratio (ζ) is an important factor in the behaviour of the system. The system, when it is influenced by a damper, can be categorized in three categories.

1. The overdamped system, where the damping ratio $\zeta > 1$. The system will return to its steady state without oscillating. When the value for the damping ratio increases, the system will need more time to return to the steady state.
2. The critically damped system, where the damping ratio $\zeta = 1$. The system will return to the steady state as quickly as possible, without oscillating. However, there might occur an overshoot.
3. The underdamped system, where the damping ratio $\zeta < 1$. The system will oscillate the amplitude that gradually goes to zero. The angular frequency of the under-damped system is given by:

$$\omega_1 = \omega_0\sqrt{1-\zeta^2} \tag{7}$$

The exponential decay of the underdamped oscillator is given by $\lambda = \omega_0\zeta$ (8)

The three categories of damping are shown in figure 25. This figure also contains an ongoing undamped oscillation.

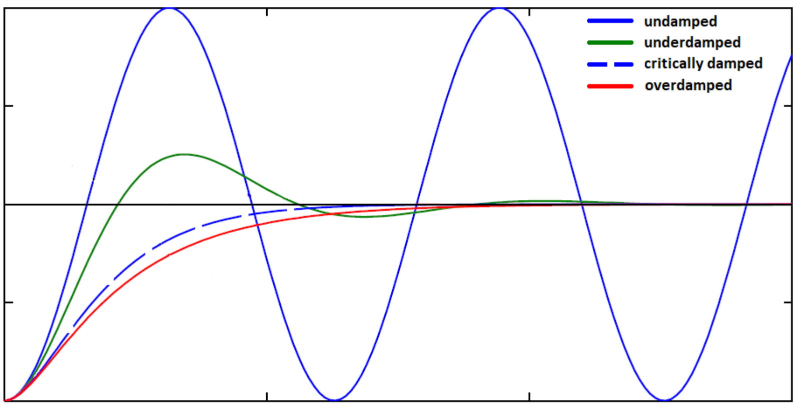


Figure 25: Three types of damping systems[15]

4.3 Check valves

The pumping system of the multi-piston pump consists of three check valves, one for each piston system. These check valves are based on a ball valve principle, as was already mentioned in chapter 2.3.1. Each check valve consists of various ball valves, to enable the flow from the pipes to the piston. These valve balls each have their specifics and characteristics. In this chapter, the physics of the ball valve mechanism will be explained. However, the design of the check valve in the ocean grazer is different than other widely used check valves, which should be taken into account.

The principle of the ball valve is that at a certain pressure, the ball moves upwards to enable a flow of the working fluid. The pressure at which the ball moves upwards in its casing is called the cracking pressure of the ball valve. The ball moves more upwards with an increasing pressure, until the maximum pressure is met. At the maximum pressure the ball valve is fully opened. The pressure and opening area of the valve can be plotted in a graph, which is shown in figure 26[16].

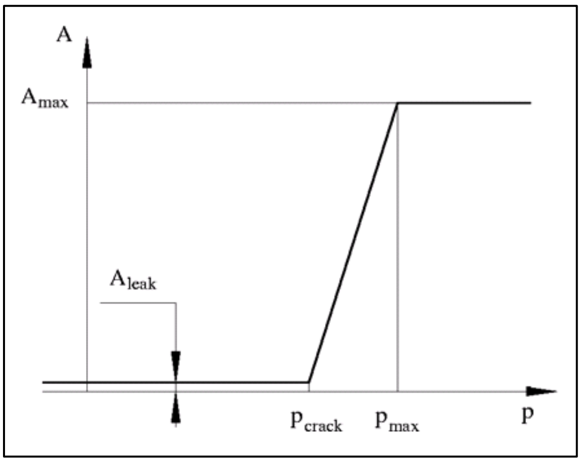


Figure 26: Area – pressure plot for valves[16]

As can be seen from the plot of the opening area and the pressure, there is always a small area which stays open. This area is called the leakage area. The relation between the opening area and the pressure can be described as follows[16]:

$$A(p) = \begin{cases} A_{leak} & for\ p \leq p_{crack} \\ A_{leak} + k(p - p_{crack}) & for\ p_{crack} < p < p_{max} \\ A_{max} & for\ p \geq p_{max} \end{cases} \quad (9)$$

This means that when the pressure is smaller than the cracking pressure, the valve is closed and only the leakage area will enable a small flow. When the cracking pressure of the ball valve is met, the ball will, as already mentioned, move up in its casing. The movement of the ball will enable the flow through the check valve. The motion of the ball valve will continue until the pressure meets the maximum pressure, where the ball valve will have its maximum opening area.

When the pressure is even higher than the maximum pressure, the ball valve may break down. The extra pressure will result in a force which will pull the valve ball to its casing, which may deform the ball. If the ball is deformed too much, the ball might not close the valve as good as it did before the deformation.

4.4 Fluid Mechanics

The experimental setup of the multi-piston pump system consists of multiple pipelines to let the working fluid flow from the lower reservoir to the upper reservoir. When using these pipelines, certain fluid mechanics have to be taken into account when designing the pipelines. In this chapter, several basic principles of fluid mechanics are explained.

The first principle that is explained is the volumetric flow rate. This flow rate defines the amount of volume that passes through a surface per unit of time (m^3/s)[17]. The formula for calculating the volumetric flow rate is as follows:

$$Q = vA \tag{10}$$

where Q is the volumetric flow rate (m^3/s), v the velocity of the volume (m/s) and A the area (m^2). The volumetric flow rate is mostly used to measure how fast a fluid travels through a pipeline.

Flow through a pipe can correspond to two types of flow, laminar flow and turbulent flow. A laminar flow is a flow which is smooth and slow. A turbulent flow is a flow that is quite chaotic and fast. The point at which a laminar flow turns into a turbulent flow is when the Reynolds number reaches a certain value, which depends on the type of fluid. The Reynolds number can be calculated with the following formula:

$$Re = (\rho v d)/\mu \tag{11}$$

where ρ is the density of the fluid, v the velocity of the fluid, d the diameter of the tube and where μ is the viscosity of the fluid. Besides the differences in velocity, laminar flow and turbulent flow also differ in energy. Due to the lower velocity, laminar flow has a lower kinetic energy than turbulent flow.

In case of the piston, some other fluid mechanic principles are involved. When the piston is in its initial position, two forces are acting on the piston. These forces are the gravitational force and the buoyancy force. The first force is calculated with the gravitational constant of the earth (9.81 m/s^2) and the mass of the piston. In formula form this leads to the following:

$$F_{grav} = mg \tag{12}$$

The second force that is acting on the piston is the buoyancy force. The buoyancy force is the force that acts on a body to make it float in a fluid. The surface of the object which is submerged is pressed back up to the surface by the fluid around it. This pressure is part of the buoyancy force. The buoyancy force that the object experiences is *"equal to the weight of the liquid displaced by the body"*[17]. The formula for the buoyancy force is the following:

$$F_{buoy} = \rho_{liq}\, g\, V_{object} \tag{13}$$

where F_{buoy} equals the buoyancy force, the ρ_{liq} the density of the fluid, the g the gravitational constant and V_{object} equals the volume of the object.

The two mentioned forces, buoyancy and gravitational, are both acting on the piston when it is both in initial position and when it is moving. The buoyancy force is acting in the opposite direction than the gravitational force. However, when an object is moving in a fluid, another force is applied to the object. This force is called the drag force and is applied by the surrounding fluid. This drag force can be seen as a kind of friction force of the movement in a fluid. In general, the formula for the drag force is the following[17]:

$$F_{drag} = C_D A_{object} \frac{\rho_{object}(v^2)}{2} \tag{14}$$

where C_D equals the drag coefficient, A_{object} the area of the object, ρ_{object} the density of the object and v equals the speed of the object in the fluid. The values for the drag coefficient are varying across the different shapes that an object can have. Even within the same shape category the value for the drag coefficient can vary. Since the drag coefficient will apply to the piston of the system, the shape category that has to be used is the cylinder shape. Besides, the direction of the flow is important as well. In this case, the flow direction should be the same as the length direction. The table of drag coefficients for cylinders with the same direction as the flow are shown in figure 27[17].

Body	**Dimensional ratio**	**Datum area, A**	**Drag coefficient, C_D**
Cylinder (flow direction)	$l/d = 1$		0.91
	2		0.85
	4	$\frac{\pi}{4} d^2$	0.87
	7		0.99

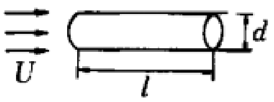


Figure 27: Drag coefficients for cylinders[17]

5. Designing the MPP model

5.1 Introduction

As mentioned before, the current model of the SPP system has to be adjusted to a functional MPP system model. To come up with an MPP model, a few blocks of the SPP model have to be adjusted. Blocks like the *Variable Head Tank*, for example, are the same for the MPP as they are for the SPP. The main difference between the two concepts, is the piston section. Therefore, the most adjustments will be done in this block.

5.2 MPP model parameters

Since the new model will introduce not only one but three pistons in the model, there are many parameters that will be added to the system as well. Not only do the pistons differ in size, the piston valve, the check valve and several pipes will also differ. Since the Simulink model and the Matlab scripts are connected to each other, both have to be adjusted. In this section, the parameters that will be adjusted are dealt with. The experimental setup, on which the current model is based, already contains the pipelines of the three pistons. However, there is only one piston which is currently used. The other two are not yet designed. Consequently, these parameters have to be assumed. To be able to make a good assumption, these parameters are calculated by using ratios of the working piston system.

5.2.1 Parameters of inflow pipes

Before the working fluid flows through the check valve, it has to flow through several pipes. These pipes are connecting the lower reservoir and the check valve. The working fluid can follow three paths, one for each piston system. The three paths are shown in figure 28. The green path flows to piston 1, the red path flows to piston 2 and the blue path flows to piston 3.

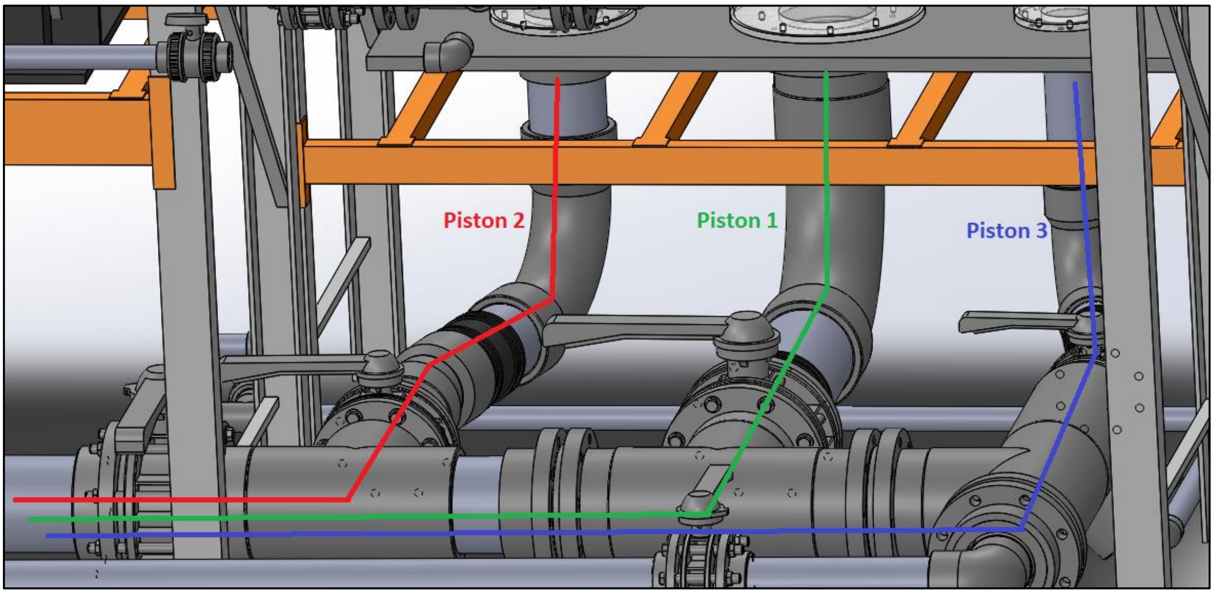


Figure 28: The inflow pipelines

These three paths have to be modelled in the Simulink model. To model them, the important parameters of the pipes have been extracted from Solidworks. One of the pipes is extracted from the Matlab code of R. Zaharia. These are shown in the following tables. Each table shows the parameters of the pipes that are connected to each other in this path. The pipe names are the same names that are used in the Solidworks model. The bend radius is the radius of the centreline of the pipe, whereas the diameters are the inner diameters. In appendix 7 a technical drawing and Bill of Material are given, to give an overview of the names and the place of the pipes.

Table 2: Inflow for piston 1

Pipe name	Diameter [m]	Length [m]	Height A [m]	Height B [m]	Bend angle [°]	Bend radius [m]
2_01_200	0.200	0.421	0.244	0.244	–	–
DN 200x26	0.190	0.260	0.244	0.244	–	–
Turn in 2_1_200	0.180	–	0.244	0.244	90	0.18
DN 200x66	0.190	0.660	0.244	0.244	–	–
1_05_200	0.160	–	0.244	0.693	90	0.293
Coupling to CV	0.200	0.146	0.693	0.839	–	–

Table 3: Inflow for piston 2

Pipe name	Diameter [m]	Length [m]	Height A [m]	Height B [m]	Bend angle [°]	Bend radius [m]
Turn in 2_1_200	0.180	–	0.244	0.244	90	0.180
DN 160x26	0.150	0.260	0.244	0.244	–	–
1_14_160	0.124	–	0.244	0.244	15	0.316
DN 160x40	0.150	0.400	0.244	0.244	–	–
1_05_160	0.124	–	0.244	0.505	90	0.230
DN 160x26	0.150	0.260	0.505	0.765	–	–
Coupling to CV	0.150	0.074	0.765	0.839	–	–

Table 4: Inflow for piston 3

Pipe name	Diameter [m]	Length [m]	Height A [m]	Height B [m]	Bend angle [°]	Bend radius [m]
2_01_200	0.200	0.421	0.244	0.244	–	–
DN 200x26	0.190	0.260	0.244	0.244	–	–
2_01_200	0.200	0.421	0.244	0.244	–	–
Turn in 2_1_200	0.180	–	0.244	0.244	90	0.180
2_01_200	0.200	0.421	0.244	0.244	–	–
1_14_110	0.086	–	0.244	0.244	15	0.253
DN 110x35	0.100	0.350	0.244	0.244	–	–
1_05_110	0.086	–	0.244	0.419	90	0.154
DN 110x35	0.100	0.350	0.419	0.769	–	–
Coupling to CV	0.100	0.070	0.769	0.839	–	–

5.2.2 Check valve parameters

The experimental setup currently contains only one check valve (corresponding to piston 1). The other check valves are not yet designed. However, the new model should contain three check valves. To model the two other check valve, several parameters must be calculated, whereas other parameters are already known. Table 5 contains the parameters that are already known. The value of the check valve pipe, both upper part and down part, takes the height of the sealing rings into account as well. These sealing rings each have a height of 3mm.

Table 5: Check valve parameters

Parameter	Piston 1	Piston 2	Piston 3
Check valve pipe diameter [m]	0.384	0.284	0.200
Check valve pipe height upper part [m]	0.318	0.318	0.318
Check valve pipe height down part [m]	0.318	0.318	0.318
Check valve height [m]	0.030	0.030	0.030
Check valve radius [m]	0.192	0.142	0.100

In this chapter the determination of the number of ball valves is approached in two ways. After the calculation, tables are given with the parameters corresponding to the check valve. To have an ideal flow in the pipes, the flow in the pipes above the check valve should be the same as the flow through the check valve. When the assumption is made that the velocities in both the check valve and the pipe are the same. One can use the formula for the flow rate to calculate the number of ball valves. This calculation is shown below for check valve 1.

Calculation check valve 1

The flow rate formula:

$$Q = vA$$

As explained, to get an ideal flow, the flow in the check valve and the flow in the pipe must be the same. Where the assumption is made that the velocity stays the same.

$$Q_{CV} = Q_{pipe}$$
$$v_{CV}\,A_{CV} = v_{pipe}\,A_{pipe}$$
$$A_{CV} = A_{pipe}$$

The area of the check valve is determined by the number of ball vales that are present. Using the formula for the area of a circle, the formula looks like the following. Where r_{cv} denotes the radius of the ball valves.

$$\sum(\pi\,(r_{CV})^2) = \pi\,(r_{pipe})^2$$

The sum sign indicates that the areas of the ball valves must be summed. One can replace the sum with the letter x to indicate the number of ball valves. The diameter of the ball valve and the diameter of the pipe are known. The radius of the ball valve is 15 mm, whereas the radius of the pipe is 95 mm. Filling in these values, the formula will lead to the number of ball valves (x) that have to be placed for the ideal flow.

$$x(\pi(r_{CV})^2) = \pi\,(r_{pipe})^2$$

The theoretical number of ball valves that have to be placed in check valve 1 is 40.15. When using the same steps for check valve 2 and 3, it follows that the values for 2 and 3 are 25.03 ball valves and 11.24 ball valves respectively. However, in the current used setting, check valve 1 has only 12 ball valves. This means that the theoretical values are not feasible in this setting. To calculate the number of ball valves for check valve 2 and 3 which are feasible in the design, one can use the factor between the area of the pipe and the area of the 12 ball valves in check valve 1. The radius of the valve balls is 0.015 m.

$$Total\ Area\ Ball\ Valves = 12\ \ (\pi\,(r_{CV})^2)$$

The area of the pipe was already calculated before, with a radius of 0.095 m.

$$Area\ pipe = \ \pi\,(r_{pipe})^2$$

The pipe Area of the pipe is 0.02835 m^2. The factor between the areas can accordingly be calculated as follows.

$$Area\ factor = Area\ pipe/Area\ Ball\ Valve$$

The factor between the areas is 3.34. With this factor, the numbers of ball valves for the other check valves can be calculated. This can be done in the following manner.

For check valve 2, with a radius of 0.075 m.

$$Area\ pipe = \pi\,(r_{pipe})^2$$

The corresponding value of the pipe area is 0.0176 m^2. Applying the factor to calculate the number of ball valves.

$$Total\ Area\ Ball\ Valves = \ Area\ pipe/Area\ factor$$
$$\#\ Ball\ Valves = \ Total\ Area\ Ball\ Valves/Area\ ball\ valve$$

The number of ball valves of check valve 2 is 7.49. When doing the same steps for check valve 3, one finds that the number of ball valves equals 3.33. Since one cannot apply 7.49 or 3.33 ball valves, the numbers are rounded upwards. This means that for check valve 2 there are 8 ball valves and for check valve 3 there are 4 ball valves. To ensure that the number of ball valves will fit in the check valve area, a sketch is made in Solidworks. Both validations can be seen in figure 29.A and figure 29.B.

As can be seen from the figure, both check valves have a configuration which fits the area of the check valve. The complete set of parameters for the check valve can be found in Table 6 (on page 39).

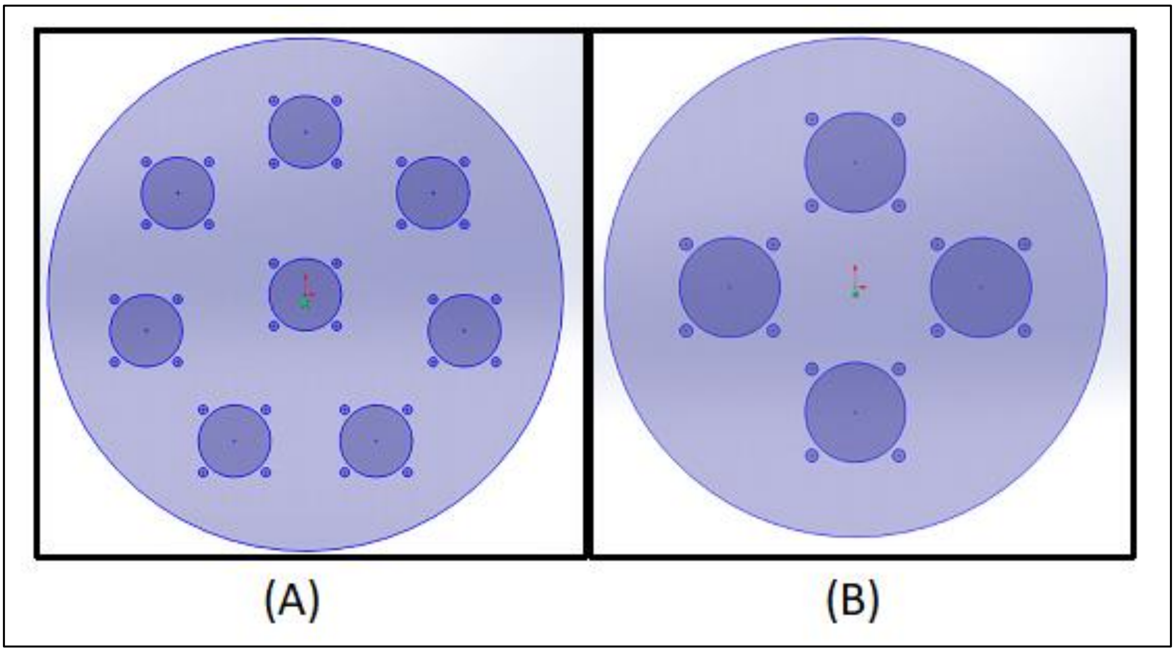


Figure 29.A: Check valve 2 configuration **Figure 29.B**: Check valve 3 configuration

Table 6: Valve parameters

Parameter	Piston 1	Piston 2	Piston 3
Radius passage area check valve [m]	0.015	0.015	0.015
Cracking pressure [Pa]	0.1 e1	0.1 e1	0.1 e1
Max opening pres-sure [Pa]	2.2 e4	2.2 e4	2.2 e4
Leakage area [m^2]	1 e-10	1 e-10	1 e-10
Opening time con-stant [s]	0.1	0.1	0.1
Initial area [m^2]	1 e-10	1 e-10	1 e-10
Number of valve balls	12	8	4

The table shows that several parameters stay the same. This is due to the assumption that is made involving the balls. It is assumed that these balls have the same cracking pressure, max opening pressure, leakage area, opening time constant and initial area. This assumption is made due to the fact that little is known about the ball valves that are placed in the experimental setup.

5.2.3 Piston and piston valve parameters

As mentioned before, the experimental setup already consists of the three pipes for the three pistons. Therefore, the parameters of the pipes are known. The diameter which is given is the inner diameter. Since the pipes are on the same level and they are equally long, the point A and B are the same. This gives the following table (table 7). In the table, Piston 1 corresponds to the piston that is currently used and has the largest diameter. The diameters that are given, are the inside diameters.

Table 7: Piston pipe parameters

Parameter	Piston 1	Piston 2	Piston 3
Pipe piston diameter [m]	0.190	0.140	0.100
Pipe piston length [m]	1.600	1.600	1.600
Pipe piston point A [m]	2.459	2.459	2.459
Pipe piston point B [m]	4.059	4.059	4.059

With these pipe parameters, one can obtain different parameters of the piston itself. The diameter of the largest pipe is 190 mm. From the technical drawing of the existing piston, it follows that the diameter of the piston itself equals 189.70 mm. This means that there is a difference in diameter of 0.3 mm. This tolerance is also used for the other two pistons. Therefore, the other two diameters can easily be calculated. The height of the pistons is assumed to be the same and can be extracted from the technical drawing as well.
As was mentioned in the previous paragraph, only one check valve is designed and working in the experimental setup. The same problem holds for the piston systems, where only the currently used piston is designed. The known values are shown in table 8.

Table 8: Known piston parameters

Parameter	Piston 1	Piston 2	Piston 3
Piston height [m]	0.1395	0.1395	0.1395
Piston diameter [m]	0.1897	0.1397	0.0997
Piston radius [m]	0.0949	0.0699	0.0499

In order to find out specific parameters of the other two pistons, calculations have to be done. The calculation involves the masses of the piston and the number of valves that have to be placed. The pistons have to be designed in such a way that they have enough time to sink, in the downstroke of the system, to the initial position. The calculation is shown below, and the results can be found in table 9.

Calculation piston masses and piston valves

Before the calculation, a plot from the old model is needed. The plot that is needed is the position plot. From this plot, the time of the downstroke can be extracted as well as the maximum displacement. The displacement plot is shown in figure 30.

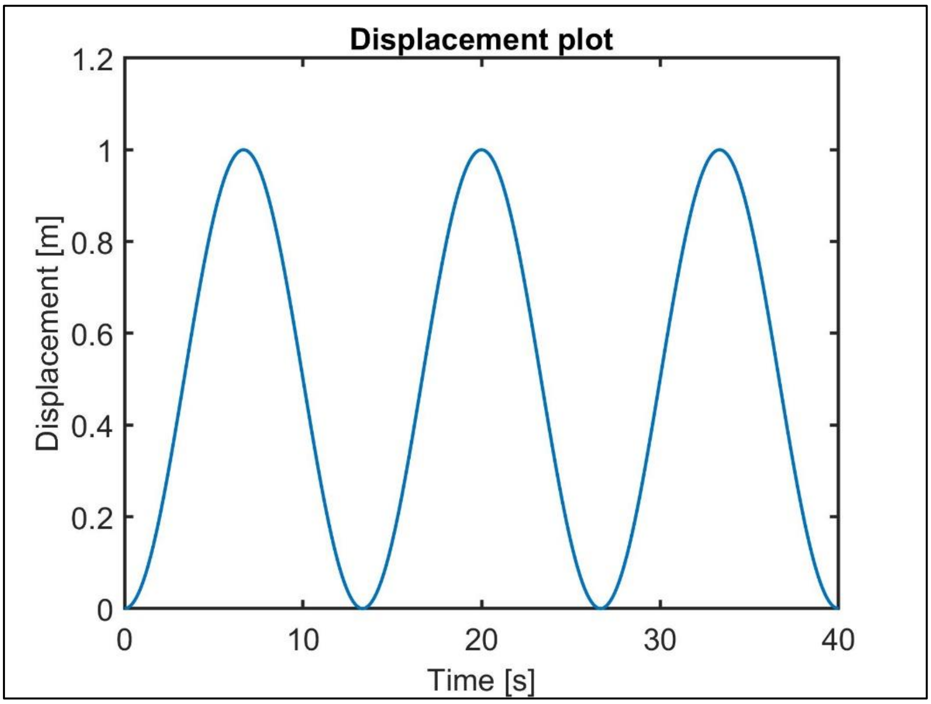


Figure 30: Displacement plot

From this plot, the time of the downstroke can be extracted. This is done in Matlab by finding the time value of the maximum displacement and the minimum displacement. The value at the maximum displacement of the first stroke equals 6,667 seconds, whereas the time of the minimum displacement equals 13.3399 seconds. Therefore, the time that one downstroke takes equals 6,67 seconds.

From the same plot, it holds that the maximum displacement of the piston equals 1 meter. Therefore, the speed equals:

$$velocity = displacement/time$$

With the downstroke time and the velocity (0.14985 m/s), one can calculate the masses and number of valves by making use of the buoyancy, drag and gravitational forces. The buoyancy and drag force both work in the same direction, whereas the gravitational force is working in the opposite direction. Therefore, one can set up the following force balance:

$$F_{buoy} + F_{drag} = F_{grav}$$

Where the buoyancy force is determined by:

$$F_{buoyancy} = \rho\, g\, V$$

And the drag force is determined by:

$$F_{drag} = C_D A \frac{\rho(v^2)}{2}$$

Using these formula, one can calculate the weight of the piston and the number of valves that have to be placed. This can be done in five steps. The calculation is shown for piston 1.

Step 1) Calculating the drag force.

$C_D = 0.91$ See figure from chapter 4.4.
$A = \pi r^2$ which equals for piston 1: $\pi(0.09485)^2 = 0.02826\ m^2$
$\rho = 8000\ kg/m^3\ (316L\ steel)$
$v = 0.14985\ m/s$

$$F_{drag} = C_D A \frac{\rho(v^2)}{2}$$
$$F_{drag} = 2.3099\ N$$

Step 2) Calculating the buoyancy.

$$F_{buoyancy} = \rho\, g\, V$$

The maximum displacement (and the displaced water height) is 1 m.

$$F_{buoyancy} = \rho\, g\, 1A$$
$$F_{buoyancy} = 277.23\ N$$

Step 3) Using a force balance.

$$F_{buoy} + F_{drag} = F_{grav}$$
$$m = 28.495\ kg$$

Step 4) Calculating the mass of the piston as a cylinder.

$$mass = \rho\, h\, A$$
$$mass = 31.538\ kg$$

Step 5) Calculating the number of piston valves by using the difference in mass.

The difference in mass corresponds to the holes of the piston valve. In step 4, the total mass is calculated if the piston is a solid cylinder. However, the piston needs holes along the total height to be able to let the working fluid flow through. Therefore, the difference is caused by these holes.

$$difference\ in\ mass = 3.05\ kg$$

the mass of the holes equals:

$$\rho\left(h_{piston}(X\,(\pi r^2))\right) = 3.05$$

Where X stands for the number of holes for the valves. The radius of one valve hole equals 15 mm. Therefore, the number of ball valves that have to be placed in the piston equals 3.87. However, the number is rounded, because one cannot apply 3.87 balls. When following the same steps for piston 2 (with piston radius 0.06985 m) and piston 3 (with piston radius 0.04985 m). One assumption is made with the velocity of the formula for the drag force, this velocity is assumed to be the same for the pistons. Plugging in the values for piston 2 and piston 3, it turns out piston 2 needs 2.09 valve balls and piston 3 needs 1.07 valve balls. The weights of the pistons are 15.45 and 7.87 respectively.

When the technical drawing of piston 1 is considered, the number of valve balls that are placed on the piston equals 6. Therefore, between the theoretical number and the actual number a factor of 1.5 holds. Using this factor on the number of ball valves of the other two pistons, the following numbers were calculated. Piston 2 needs 3 ball valves and piston 3 needs 1.5 ball valves, where the last number is again rounded up. Figure 31.A and figure 31.B. are showing the configuration of Piston 2 and Piston 3.

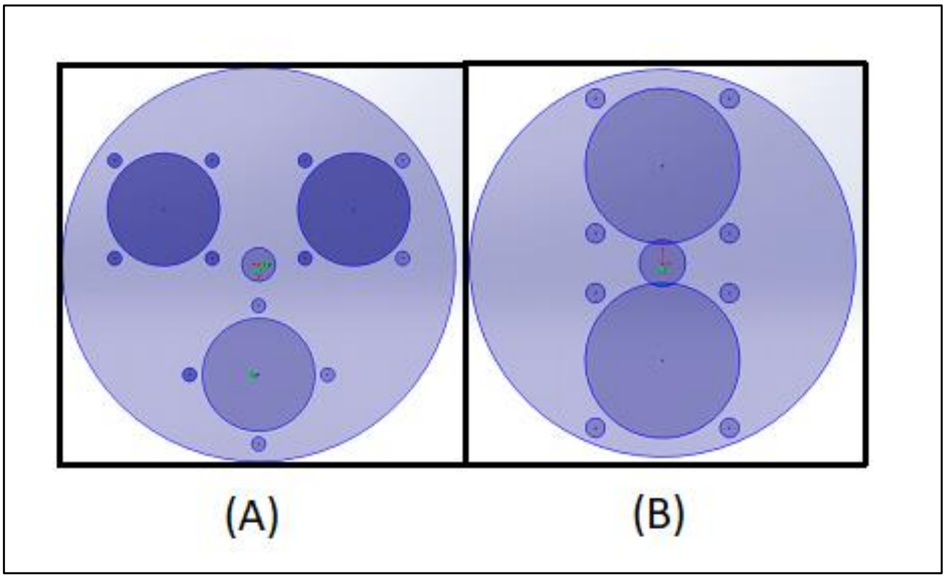


Figure 31.A: Piston 2 configuration **Figure 31.B**: Piston 3 configuration

As can be seen from the figure, the configuration of piston 2 will hold. However, the configuration of piston 3 does not hold. The holes of the valves are intersecting the hole where the cable will be connected to the piston. To obtain a configuration which would fit in the area of the piston, one can consider decreasing the radius of the valve balls. However, one major drawback is that it is not known what this decreasing does to parameters as cracking pressure or opening time constant. Therefore, the theoretical value of 2 valves is used in this research.

Table 9: Piston and piston valve parameters

Parameter	Piston 1	Piston 2	Piston 3
Piston mass [kg]	28.56	15.45	7.87
Radius passage area piston valve [m]	0.015	0.015	0.0125
Cracking pressure [Pa]	0.1 e1	0.1 e1	0.1 e1
Max opening pressure [Pa]	2.2 e4	2.2 e4	2.2 e4
Leakage area [m^2]	1 e-10	1 e-10	1 e-10
Opening time constant [s]	0.1	0.1	0.1
Initial area [m^2]	1 e-10	1 e-10	1 e-10
Number of valve balls	6	3	2

5.2.4 Other important parameters

At last, there are a few parameters that are not yet dealt with but have to be taken into account as well. Several of these parameters are directly extracted from Solidworks, while others may be extracted from the Matlab code of R. Zaharia[14]. The values of these parameters can be found in table 10. In this table the parameter *Length pipe CV-Piston* is mentioned. This parameter is corresponding to the pipe between the check valve casing and the piston casing. The same holds for the parameter *Diameter pipe CV-Piston*.

Table 10: Other parameters

Parameter	Piston 1	Piston 2	Piston 3
Length pipe CV-Piston [m]	0.540	0.540	0.540
Diameter pipe CV-Piston [m]	0.190	0.150	0.100
Cable length [m]	5.0	5.0	5.0
Cable diameter [m]	0.005	0.005	0.005
Cable Youngs Modulus [MPa]	210 10^3	210 10^3	210 10^3

5.3 The adjusted model

In this chapter the adjusted model is described. The previous calculated parameters have to be added to the Matlab script *initialization_parameters.m*. After that, the Simulink model can be adjusted to an MPP model. The different parameters must have a different name, this name is linked to the Simulink model. Therefore, when different parameters have the same name, the model will not work.

Before the model is described, it is important to mention one assumption that is made during the modelling of the MPP system. The assumption stands in direct relation to the ball valves of both the check valves and the piston valves. Due to the fact that the piston valves and check valves in the system are custom designed, it is difficult to scale these parameters in the right manner. The assumption, which is mentioned in both paragraph 5.2.2 and paragraph 5.2.3, is that these parameters stay the same.

As said in section 5.2.3, the configuration of the valves in piston three is not possible. Therefore, it was checked if a configuration with smaller balls would fit. The conclusion as that valve balls of 12.5 mm would fit in the piston. In the model, the configuration of piston 3 involves two piston valves. This decision is based on the assumption that was made concerning the parameters of the valve itself. When scaling down the piston valve balls, it is not known what the scaling will be for, for example, the cracking pressure or the opening time constant.

Before modelling the MPP system in Simulink, it is useful to find the related outputs of the model. These outputs are generated by the Simulink model and are transferred to the workspace of Matlab. To obtain this information about the three pistons of the MPP system, one has to take into account that these outputs must have distinct names. Therefore, the parameters are renamed according to the corresponding piston (table 11). For example, the output of *piston_pos1* is renamed to *piston_1_pos, piston_2_pos and piston_3_pos*. The same holds for the parameters, from the initialization script, that are connected to the modelling blocks. All these parameters are renamed in the same way as the outputs of the model.

Table 11: Changed outputs of the model

Output old model	Piston 1	Piston 2	Piston 3
p_cv	p_cv1	p_cv2	p_cv3
flow_cv_total	flow_cv1_total	flow_cv2_total	flow_cv3_total
piston_pos1	piston_1_pos	piston_2_pos	piston_3_pos
piston_velocity	piston_1_velocity	piston_2_velocity	piston_3_velocity
motor_force1	motor_force1	motor_force2	motor_force3
aFrequency	aFrequency1	aFrequency2	aFrequency3
p_pv	p_pv1	p_pv2	p_pv3
flow_pv_total	flow_pv1_total	flow_pv2_total	flow_pv3_total
Piston_force	piston_1_force	piston_2_force	piston_3_force

After renaming these outputs, the parameters that were discussed in the previous paragraph are changed and added into the *initialization_parameters.m* script. This script is also, for the MPP system, renamed to *initialization_parameters_MPP_V2.m* to distinguish the SPP model and the MPP model. When both the outputs and inputs (the parameters) are changed and added, the Simulink model can be adjusted.

The new model is shown in figure 32. The inflow pipelines are modelled in the piping subblock and each path is, in the new model, connected to a check valve which enables the flow to the rest of the system. Besides three new check valves, there are two added pistons and

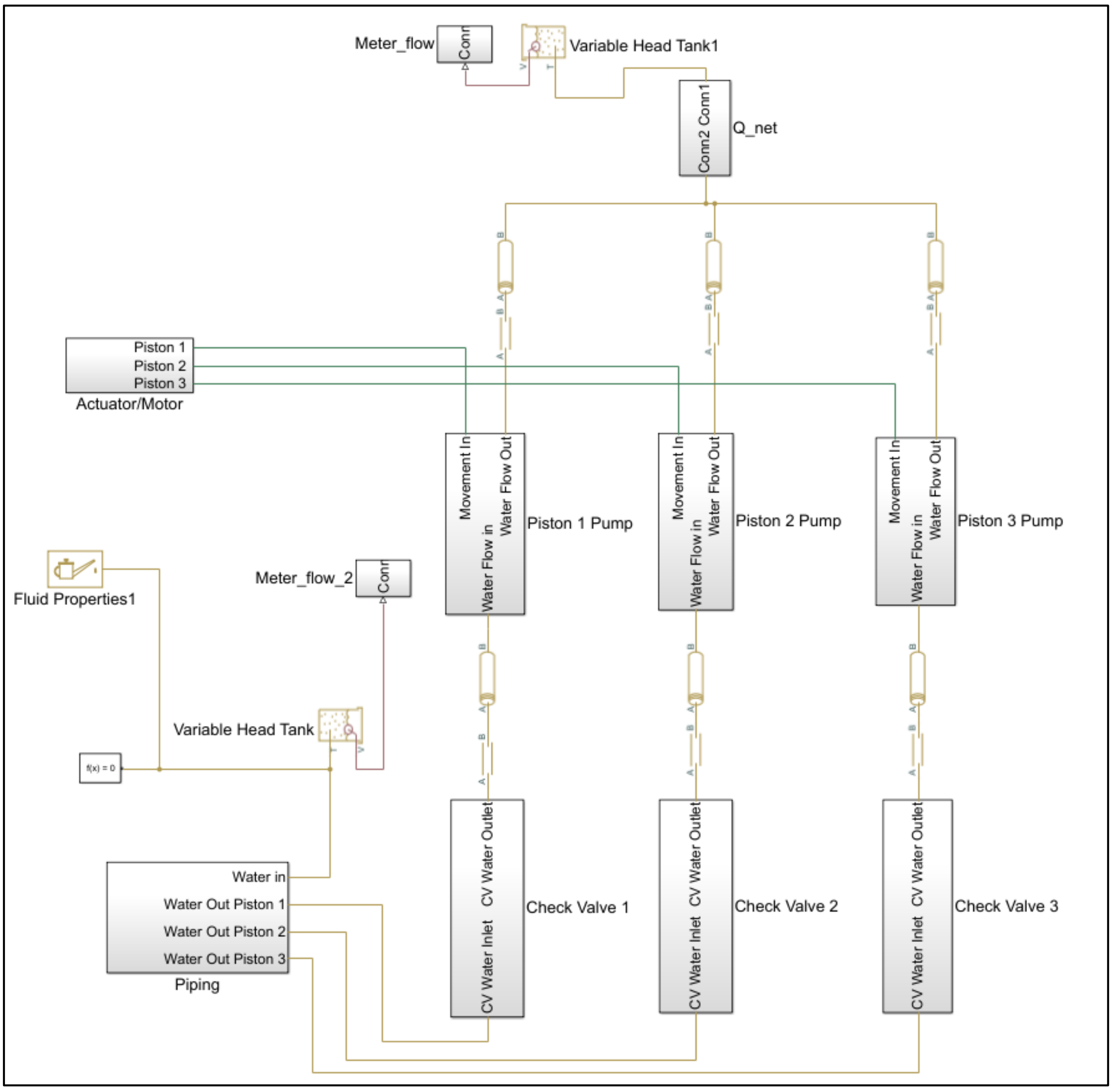


Figure 32: The new MPP Simulink block model

piston pipes. The other main difference is that the motor now is connected to three pistons, whereas in the old model the motor was connected to only one piston.

As said, the inflow pipelines are modelled in the sub-block of the piping. In this sublayer of the model, the different paths are shown as they were presented in paragraph 5.2.1, figure 33. The pipes are numbered corresponding to the path that they follow. Exceptions for this numbering are the pipes which used in multiple paths.

To be able to understand the interaction between the pistons when they are activated or de-activated, one can build an actuation mechanism. Due to the many factors that are involved in the actuation mechanism itself, it is difficult to build a complete actuation mechanism in Simulink. However, when a simple activation mechanism is considered, one can at give a prediction on what happens when different pistons are activated or deactivated. The actua-tion mechanism that was built within this research is a basic switch system, build in the same block as the motor.

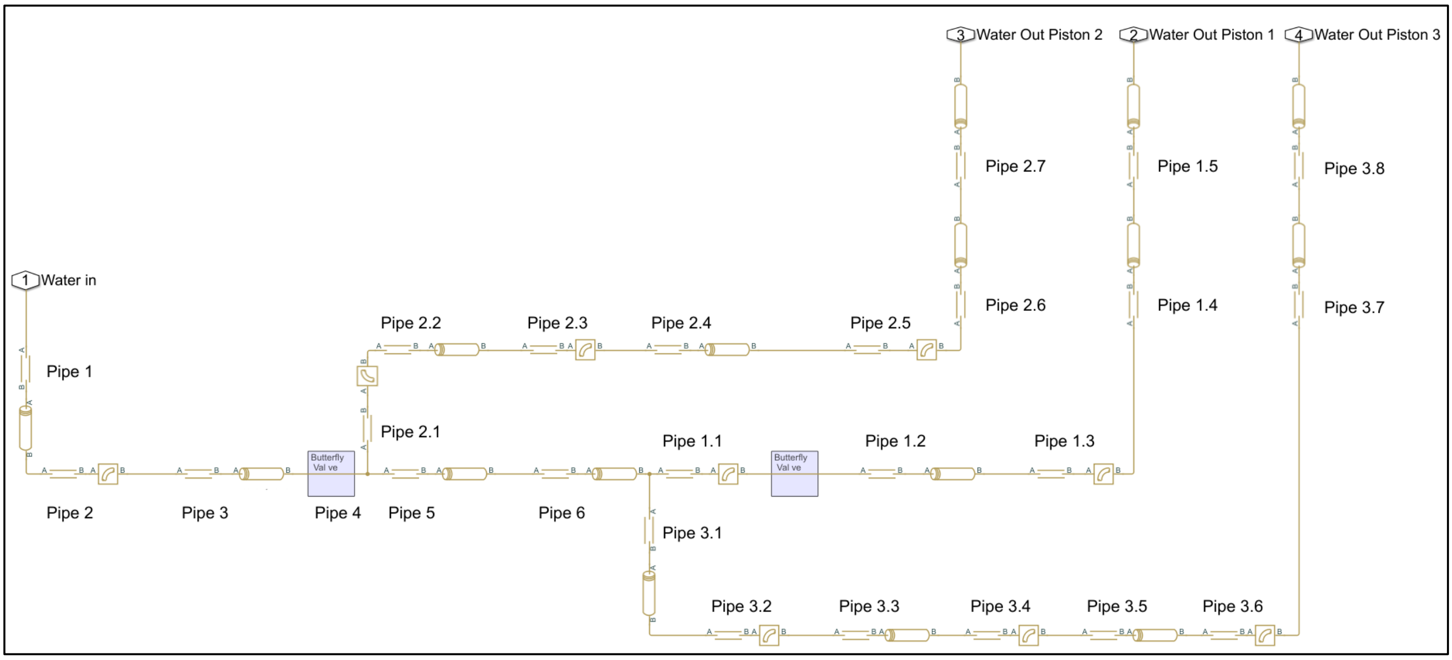


Figure 33: The inflow pipeline subsystem for the MPP model

A drawback of the activation mechanism as it is modelled now, is that it is not capable of changing piston combinations during a running simulation. This is a limitation of the software of Matlab and can possibly be modelled by doing research in the actuator mechanism itself. However, before a simulation is ran the combination of pistons can be set in variant subsystems that are used. Besides the switches, an additional spring damper system is added. This addition models the cable which is connected to the actuator and the motor. The variant subsystems that are used have two possible outputs. When it is active, the system enables the sine wave signal to pass through. When the variant subsystem is inactive, it will switch to only the reference and the piston will not be active. The outline of the actuation mechanism is shown in figure 34.

The figure shows that the actuation mechanism is modelled with three variant subsystems. As mentioned earlier, when the block is active, it enables the sine wave to pass through. If the block is inactive, the piston will only be coupled to a mechanical reference and will therefore not be activated. In this sub-model, one can see that there is an added rod, modelled by a spring damper system, which connects the motor and the actuation mechanism.

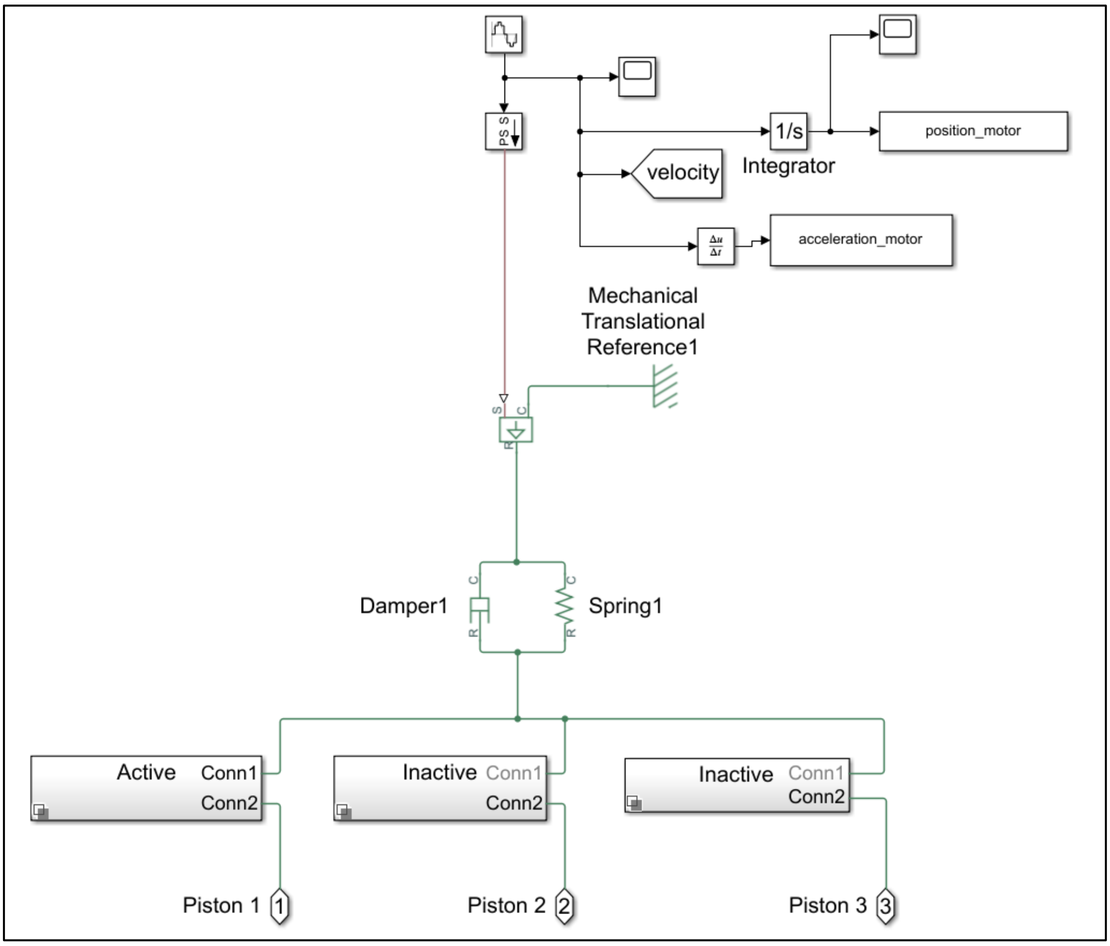


Figure 34: The new motor and actuator subsystem

6. Analysing the behaviour

The obtained model from chapter 5.3 can now be used to analyse the behaviour of the multi piston pump. The graphs that were obtained from the model are discussed in this chapter. After a simulation is done, a table with different calculated results (from *calculationsSim-ulink_MPP_V2.m*) is given by Matlab. These results, with all the pistons set to active, are shown in figure 35.

```
----------------------------------------------------------
|           Results Simulink Model            |
-------------------------
Average Max displacement = 1 m
-------------------------
Total Pumping energy = 4198.4235 J
Total Potential energy = 1091.1648 J
Avg. Pumping energy per stroke = 1399.4745 J
Avg. Potential energy per stroke = 363.7216 J
-------------------------
Volumetric Efficiency = 64.2684 %
Mechanical Efficiency = 40.4396 %
Total Efficiency = 25.9899 %
```

Figure 35: Results of the calculations script for the MPP

From the calculation script, it becomes clear that the average maximum displacement stays the same when the SPP model is built to an MPP model. The volumetric efficiency is based on the pumped fluid and the maximal fluid that could have been pumped. The total efficiency is based on the potential energy stored and the energy which is needed to pump the fluid. Table 12 shows the different efficiencies and average potential and pumping energy. However, the results that are shown are all corresponding to the same motor settings. In the real situation the combinations each have a specific range of motor settings which determine if the combination is used or not.

Table 12: Results from the Matlab script

Combination	# strokes [-]	η_{vol} [%]	η_{mech} [%]	η_{tot} [%]	avg. E_{pot} [J]	avg. E_{pump} [J]
P1	3	70.02	34.36	24.06	217.88	905.58
P2	3	85.23	32.47	27.67	143.62	518.99
P3	3	92.08	29.17	26.86	78.87	293.64
P1 & P2	3	66.14	38.58	25.52	318.07	1246.58
P1 & P3	3	69.03	36.00	24.85	274.13	1102.98
P2 & P3	3	82.75	33.29	27.54	210.43	763.99
P1 & P2 & P3	3	64.26	40.43	25.99	363.72	1399.47

However, the efficiency does not account for any losses that are due to, for example, the pulleys in the system. Because of that, the actual efficiency of the system will be lower than the simulated efficiency in this model. When the efficiencies of the model are compared with

The efficiency of the volume is influenced by two factors, a backflow and leakage. When the activated piston combination is in the upstroke, the volume below the check valve of the in-activated piston can flow back. On the other hand, in the check valve and piston valve will always occur some leakage.

6.1 Forces

The first plot that is obtained is the force plot. This is the same plot as was shown in paragraph 4.1.3. However, there are now three piston forces, one for each piston. The obtained plot is shown in figure 36.

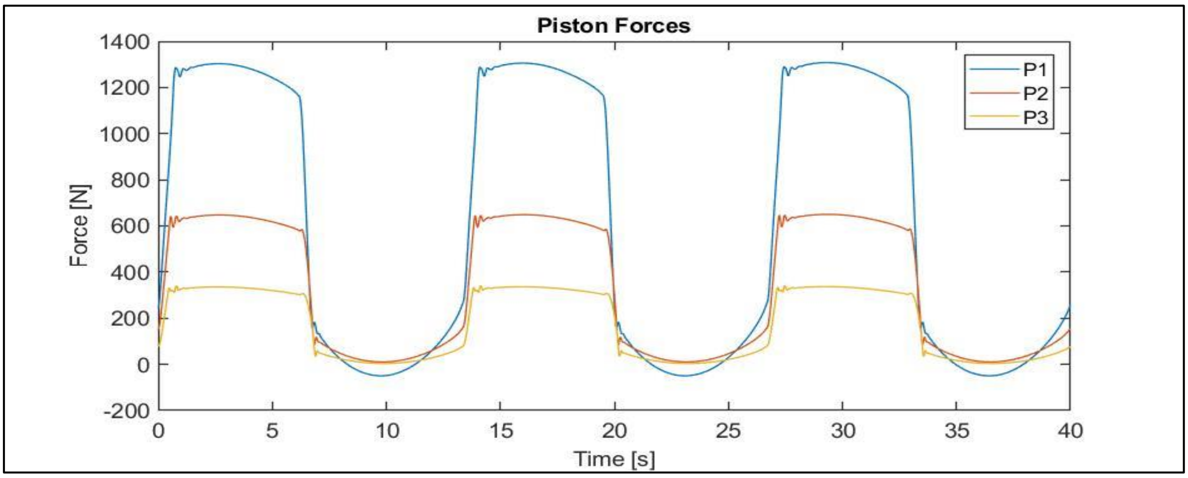


Figure 36: The piston forces in the MPP

This graph shows that the force of the piston decreases when the piston becomes smaller in the system. This behaviour makes sense due to the decreasing water column above the piston which becomes smaller as well. Besides plotting the different forces into one plot, one can plot the combined force as well. This can be done because the three forces are applied in the same direction. The combined force plot is shown in figure 37.

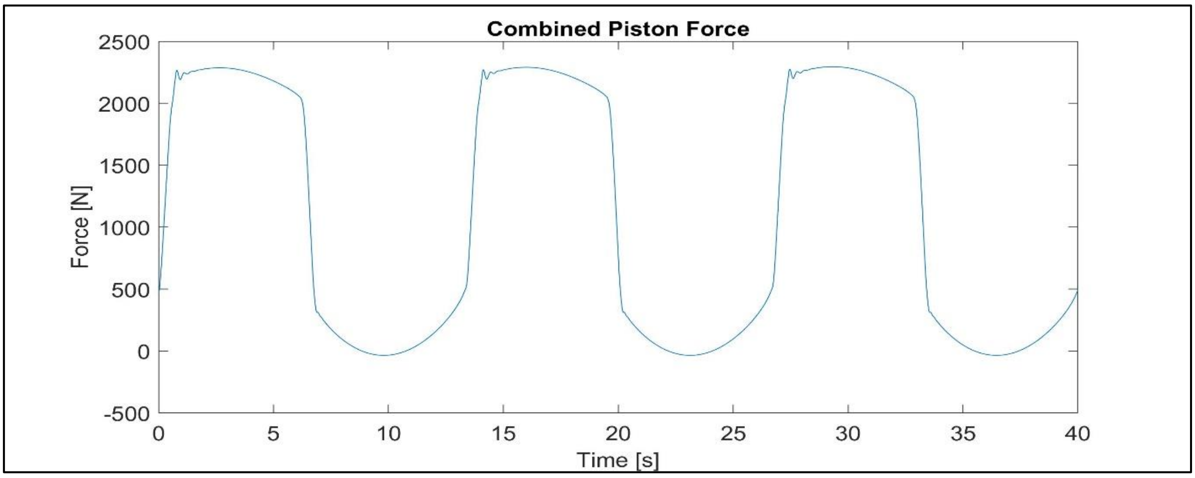


Figure 37: The combined piston force of the MPP

This combined force plot shows that the coupling mechanism, to which the pistons are attached, has to handle a combined force of more than 2000 Newtons. Extracting the maximum value of the combined force results in a value of 2356.1 Newton.

6.2 Pressures and flows

As was mentioned in chapter 4.1.3, the model is able to plot the flows over both the check valve and piston valve. The flow through both check and piston valve is enabled by the pressure which opens the ball valves. The pressure plots for the MPP model is shown in figure 38.

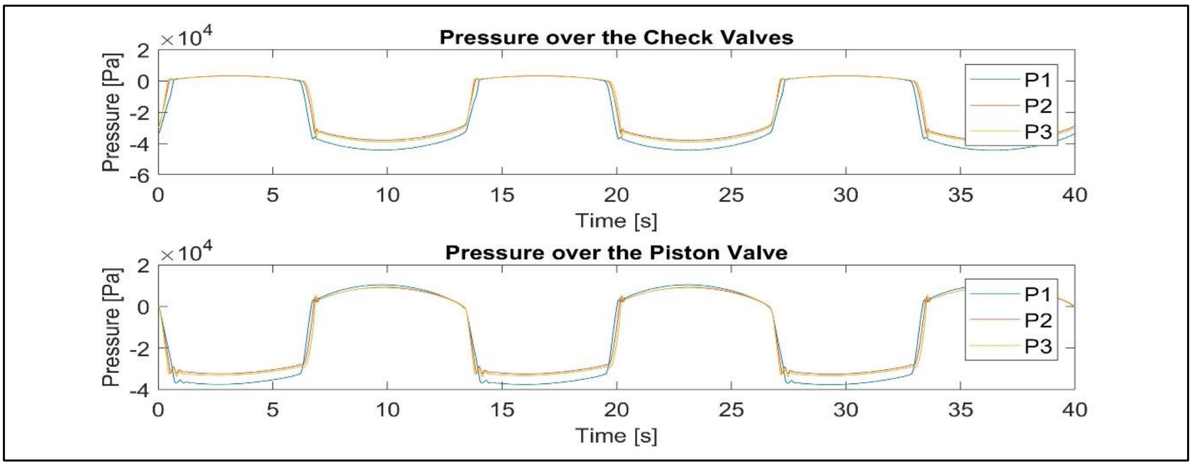


Figure 38: The pressures over the check and piston valves

The plot shows the difference in the opening of the different valves. The check valves are opened in the upstroke, hence the pressure is 0, and the piston valves are opened in the downstroke, where the pressure will be 0. The flow that these pressures enable are shown in figure 39.

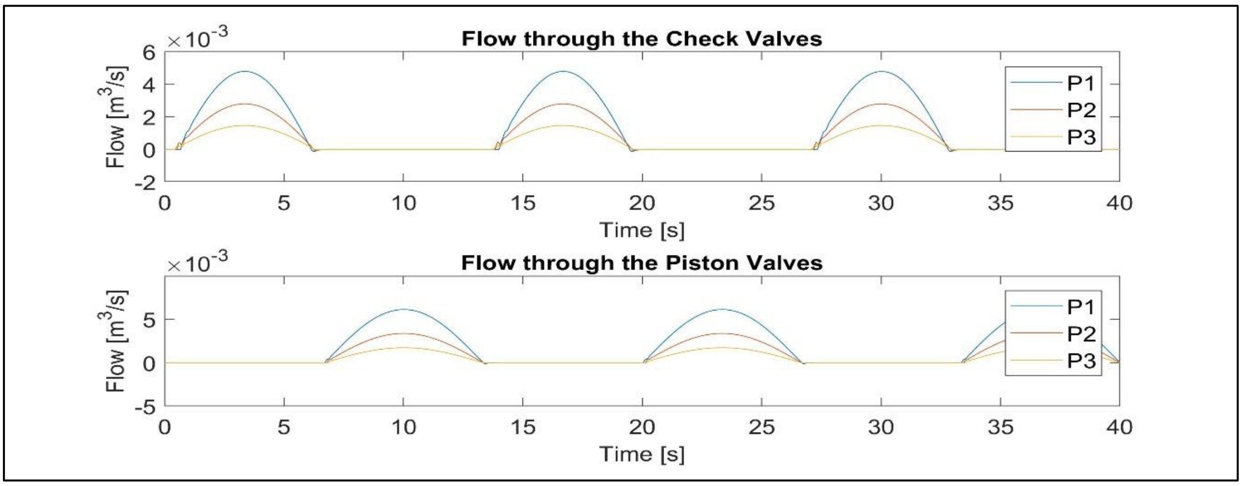


Figure 39: The flows through the check and piston valves

The graph of the flow through both valves shows the same behaviour as the pressure plot in terms of the time the valve opens. The up and downstroke of the system can be seen clearly from the graphs. During the upstroke, the flow through the check valve is enabled, whereas during the downstroke the flow through the piston valve is enabled.

6.3 Actuation mechanism

When incorporating the actuator mechanism, one can better understand the behaviour of the different piston combinations. In total, there are 7 possible combinations. Using the actuator to plot the different combinations with their corresponding combined forces over time. However, the model takes the gravitational force into account as well, whereas the actual system will not have to deal with these forces. The 7 piston combinations and their combined forces are shown in figure 40.

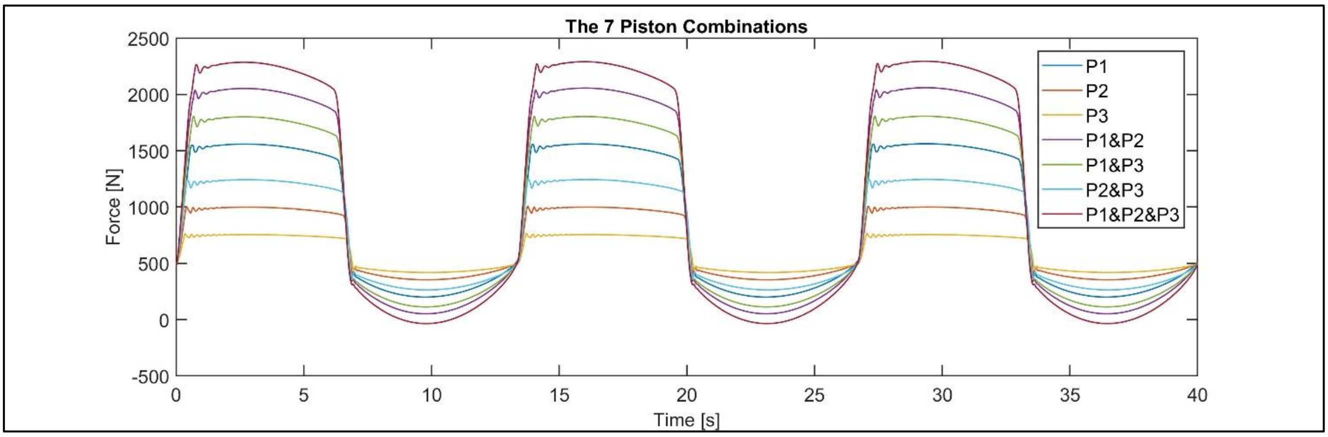


Figure 40: The combined piston forces over the 7 piston combinations

What can be seen from the graph is that as the combined force lowers, the amplitude of the oscillation of the force decreases as well in the upstroke. The graph also shows that as the total force decreases, it takes more time for the system to damp out the oscillations. In the system where all the pistons are active, the system needs three peaks in the oscillation, whereas for the combination with only the third piston active, the system needs more time. However, looking at all the oscillations, one can say that the system is underdamped in each piston combination. In figure 41 an enlarged plot is shown for the upstroke.

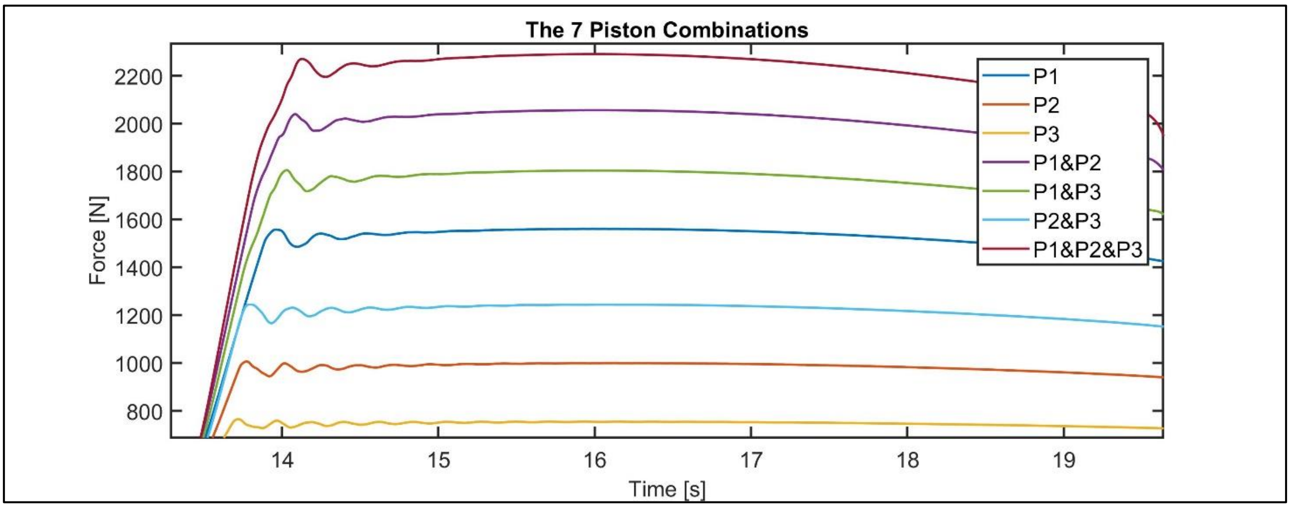


Figure 41: Combined piston forces over the 7 combinations in the upstroke

The downstroke of the system is more chaotic than the upstroke of the system. What can be seen is that the heavier the combination is, the lower the amplitudes of the downstroke oscillations. An enlarged plot of the oscillations of the downstroke can be found in figure 42

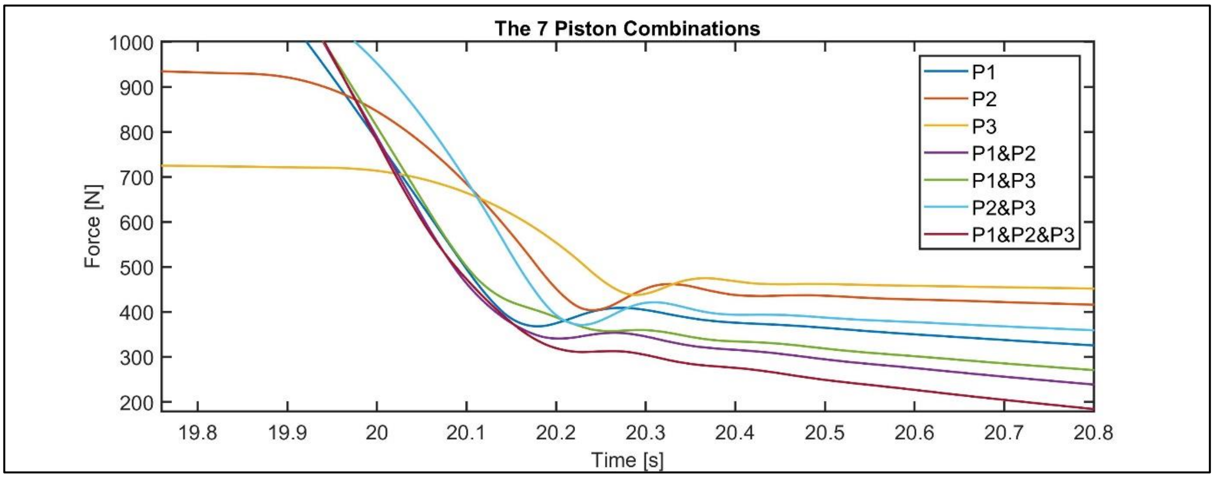


Figure 42: Combined piston forces over the 7 combinations in the downstroke

The frequency of this oscillation is together with the force an important factor in designing the actuator/coupling mechanism. This mechanism will fail when it cannot hold the combined force of the pistons or when the frequencies of the combinations will start to resonance with the natural frequency of the mechanism itself. As described in chapter 4.2, the natural frequency of the system is calculated with the following formula:

$$f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$$

The frequency of an oscillation is the number of times that a wave is produced in a period of time. In the case of the graphs in figure 41 and 42 one can say that for the upstroke the frequency increases as the total force lowers. The downstroke of the systems tend to show that as the combination becomes heavier, the frequency becomes smaller.

In the simulation, the cables are assumed to be the same. However, one can decide to place different cables with different specifics. In the simulation, the damping coefficient can be changed with respect to the piston mass. In this case the four damping coefficients of the cable are set to the following (table 13).

Table 13: Damping coefficients

	Cable 1	Cable 2	Cable 3	Cable 4
Connection	Motor – Actuator	Actuator – Piston 1	Actuator – Piston 2	Actuator – Piston 3
Damping coefficient	43000	30000	20000	10000

The values of the damping coefficient are not the actual values of the cables since these are difficult to determine. In the research of R. Zaharia, the value of 43000 was used and therefore this value will be set for the cable between the motor and the actuator. In figure 43 and figure 44 the results of these settings are shown and discussed.

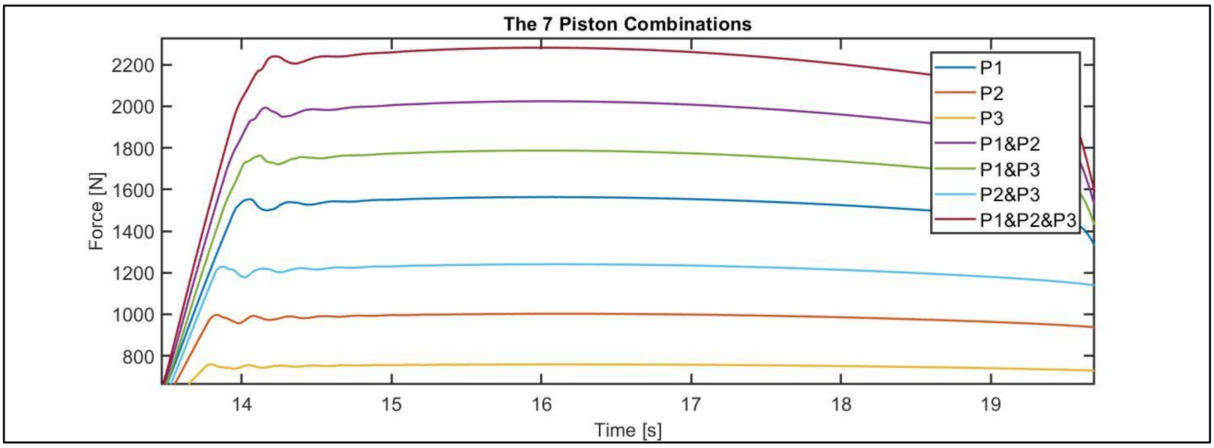


Figure 43: Combined piston forces with smaller damping in the upstroke

Considering the lower damping coefficients and figure 43, one can say that for the upstroke the oscillations have a smaller amplitude. The other point that this plot shows is that the oscillation takes less time to damp out.

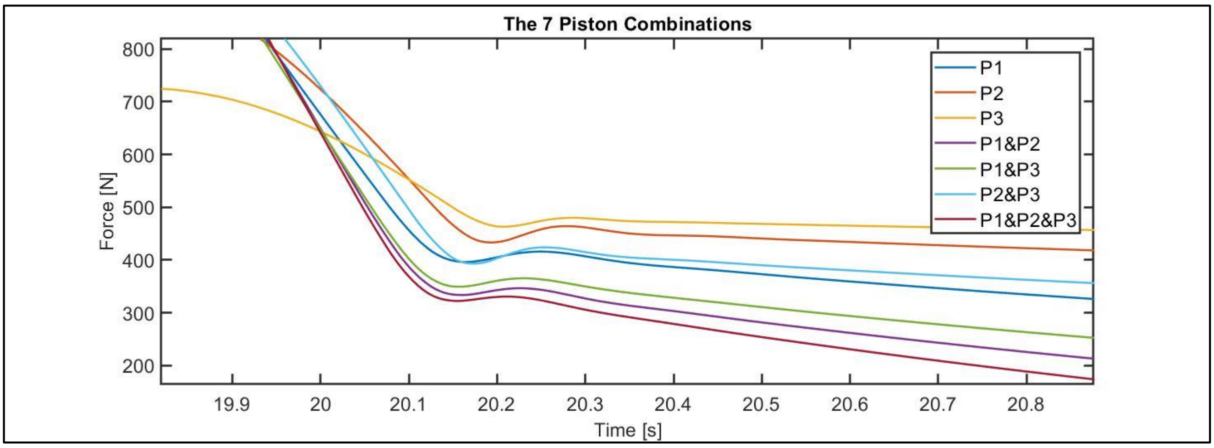


Figure 44: Combined piston forces with smaller damping in the downstroke

The graph in figure 44 shows that when the damping coefficient is set based on the mass of the piston, the oscillations in the downstroke become less chaotic. The amplitudes of the oscillations become smaller as well, which also happened in the upstroke. However, considering both figure 43 and 44 one can say that the system is still underdamped. However, there is already better damping than before. In terms of frequency, one can say that the frequency still increases when the total force decreases in the upstroke. In the downstroke one can see that as the combinations becomes heavier, the frequency increases. This behaviour in frequency was the same in the simulation where the cables were assumed to be the same.

Another interesting output to analyse is the behaviour of the different pistons in the different combinations. Where in the first part of this chapter the focus was on the combined force, the focus lies here in the individual forces on the pistons. Each piston has four combinations in which the piston is active. Table 14 shows the different combinations in which the different pistons are active.

Table 14: Active Piston in combinations

Piston	Combination 1	Combination 2	Combination 3	Combination 4
Piston 1	P1	P1 & P2	P1 & P3	P1 & P2 & P3
Piston 2	P2	P1 & P2	P2 & P3	P1 & P2 & P3
Piston 3	P3	P2 & P3	P1 & P3	P1 & P2 & P3

In the case of piston 1

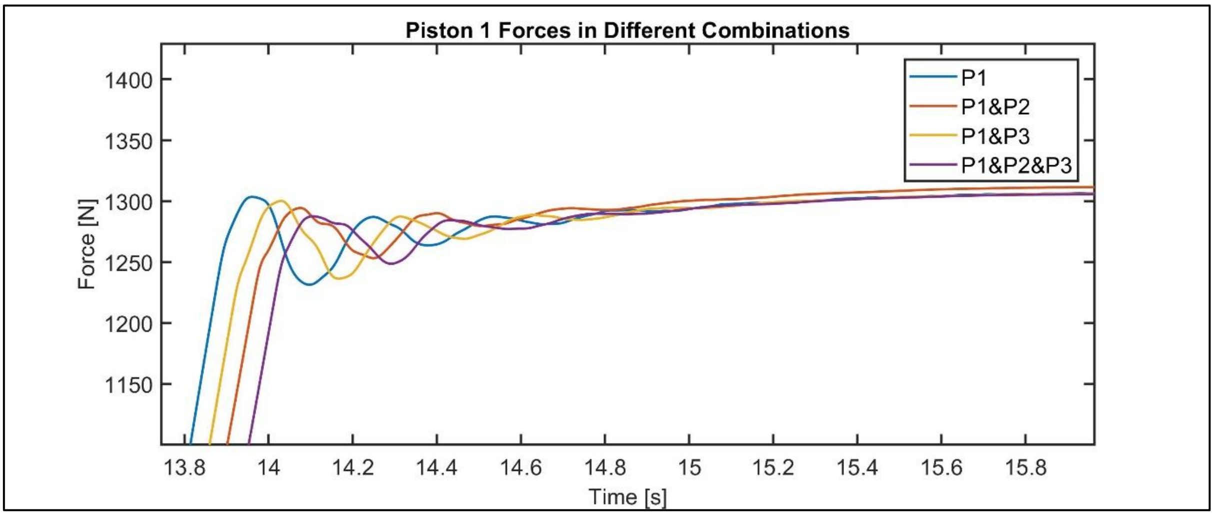


Figure 45: Forces on piston 1 over the combinations involving piston 1

In the case of piston 2

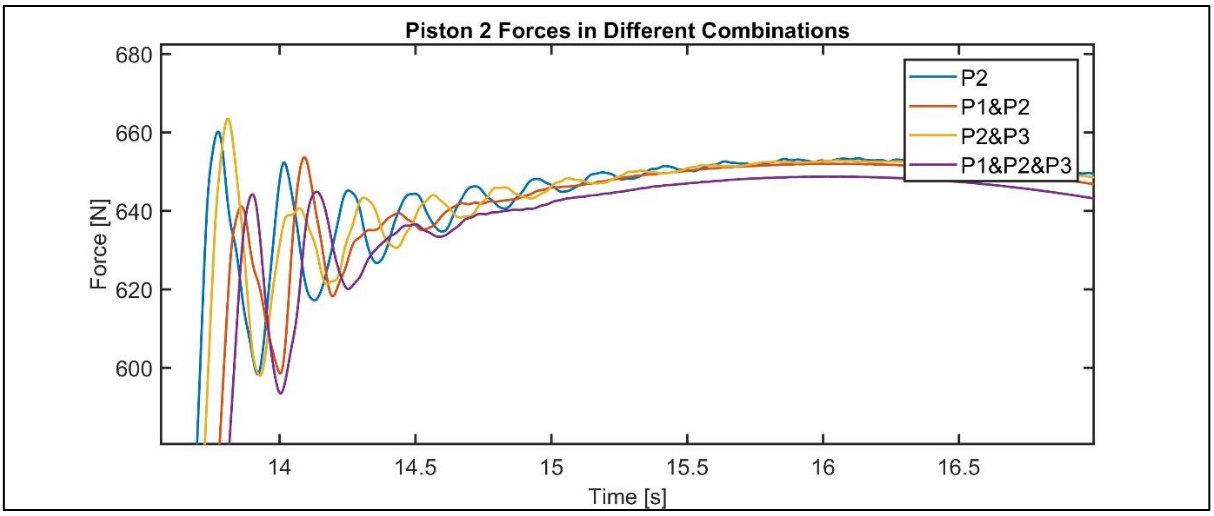


Figure 46: Forces on piston 2 over the combinations involving piston 2

In the case of piston 3

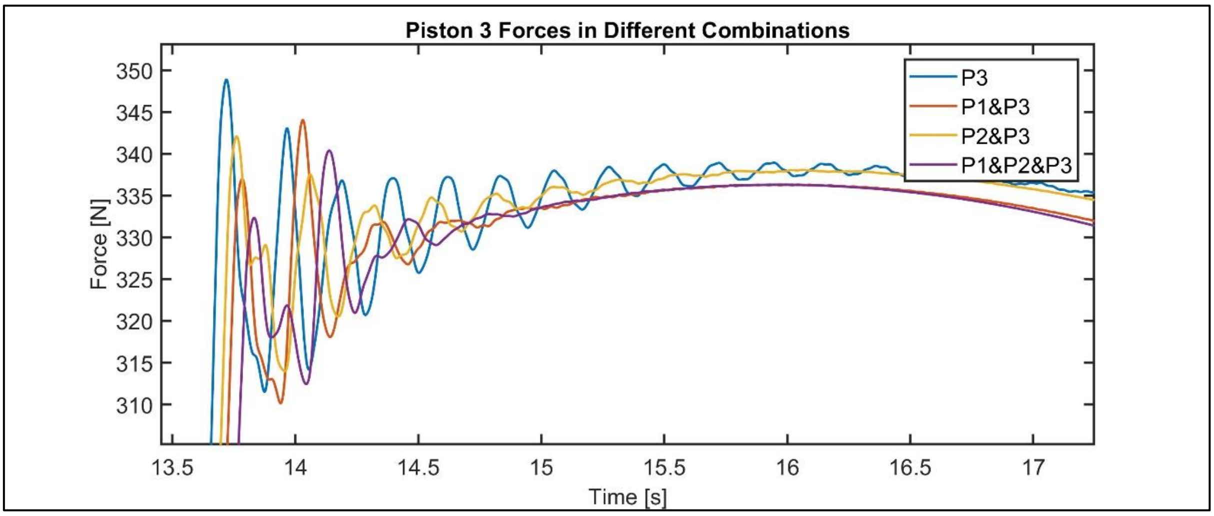


Figure 47: Forces on piston 3 over the combinations involving piston 3

The graphs of the piston forces in the combinations show that as the combination becomes heavier, i.e. the combined mass increases, the oscillation in the force has lower amplitudes and is damped out earlier. On the other hand, the lower the mass of the pistons, the longer the oscillations take to damp out and the more chaotic the graphs tend to be. The frequency of the oscillations tends to become higher when the combination is less heavy.

As previously mentioned, the actuator of the system is quite simple. Therefore, some physical phenomena might not be modelled in this model. This can lead to different outputs when the model is further improved in the actuator subsystem. Besides the actuator, the damping coefficient and the spring constant can influence the system. These numbers are not known yet for the system.

7. Conclusions

7.1 Conclusions

Based on the graphs that were derived from the MPP model, one can say that the system behaves in way that is comparable with the SPP model. The graphs show for example a decreasing flow through the check valves and piston valves with a decreasing piston size. This graph is can therefore validate the system. The graphs of both the flow and the force tend to show the ratio of 1:2:4, which was given to the system when it was designed. The calculation script provided that the efficiency of the system decreased from 30.43 to 25.99 percent. However, due to several friction factors, which are not yet accounted for, the efficiency of the system might turn out to be lower than the 25.99 percent.

From the table of the produced efficiencies one can conclude that, for the efficiencies, the model contains an error. The efficiencies that the model produced are not in line with earlier models of the MPP. When compared with these older models, the efficiencies turn out to be too low. An indicator for this is the difference in efficiency with the SPP model. The efficiency of the SPP model should be the same as the efficiency of the MPP model when only piston 1 is set to active. However, it turned out that there was a difference of around 6%. However, the SPP model that was the foundation of the MPP model also produced efficiencies that were not in line with the older models. However, the behaviour of the MPP model is comparable with the SPP model in terms of force, flow and pressure. Therefore, the model is a useful tool to predict the behaviour of the multi piston pumping system.

It can be concluded that it is possible to adjust an SPP model to an MPP model. This can be done by adding the different subsystems that are necessary for the MPP system. These subsystems and their building blocks must be assigned to their own parameters to let the model work. Parameters that are not known can be calculated and predicted by their physical phenomena. The main difference between the two models is that, almost, every output has 3 outputs, instead of 1, in the new model. On the other hand, the model can also show the interactions between the three pistons. The force output of the system can be interpreted in different ways. One can consider three force outputs: the combined piston force, the combined piston forces in the 7 combinations and the individual piston forces in the combinations.

Considering the research question in chapter 2, the force and frequency in the force output can be used to draw conclusions about the system. As already mentioned, the MPP system makes use of an actuation mechanism to couple the different pistons. The different combinations react differently to the system. Plotting the different combinations, one can see that as the total piston force lowers, the amplitude of the oscillations lowers as well. On the other hand, the oscillation takes more time to be dampened out when the total force lowers. When the damping coefficient is set accordingly to the mass of the piston and not assumed to be the same, one can conclude that oscillations will have a lower amplitude and will take less time to damp out. In both cases the frequency tends to increase when the total force decreases in the upstroke for the different combinations.

In the case of the individual forces of the pistons in the different combinations, the force oscillations have lower amplitudes when the combination becomes heavier. This can be caused by a change in the distribution when more pistons are activated. However, when in a combination the combined force is considered, the amplitude will increase when the combination becomes heavier. The oscillation is, in the case of the individual pistons, also dampened out earlier when the combination is heavier. In terms of frequency, one can conclude that when the combined mass is lower, the frequency becomes higher.

The discussed lowering of the amplitude of the oscillation and the time that it needs to damp out gives a design specification for the actuator mechanism. Firstly, the actuator mechanism has to deal with the piston forces which are acting on the actuator. If the material of the actuator is not capable of handling these forces, it will fail. The same holds for the oscillations in the system. As already explained in chapter 6, the oscillations should not resonate with the natural frequency of the actuation mechanism. When the system starts to resonate on the natural frequency of the actuation mechanism, the system will fail as well. The system should also be able to handle oscillations that might last longer than other oscillations.

One can say that the model, that was derived in this research, gives the Ocean Grazer research team a useful tool to simulate the MPP system. Although there are still some limitations in the model, it will give important insights in the behaviour as well as the energy outputs of the system.

7.2 Discussion and limitations

As said before, the new MPP model gives the Ocean Grazer project team a new tool to simulate the behaviour of the MPP. The model is capable of simulating all the seven combinations that are possible with the MPP system. However, there are some limitations concerning the simulations of the MPP.

The first limitation is that the model is not able to switch from combination during a simulation. For each simulation one combination can be simulated. This is a limitation which is in the software of Matlab and Simulink. The consequence is that it is not possible to investigate the behaviour during the switching of combinations. However, it is possible to plot the different combinations and plot them in one graph. This does give an insight in the oscillations that are occurring during the up and downstroke of the system.

A second limitation lies within the efficiencies of the system. As stated in the conclusion, the efficiencies turned out to be too low, when compared with older models. However, in terms of force, flow and pressure the model is a useful tool to analyse and predict the behaviour of the multi piston pump system. More research can be done to investigate the efficiencies of the system.

Another limitation concerns the check and piston valves. The limitation is due to the fact that the two other sets of valves are not yet designed. Therefore, the number of ball valves are calculated and in the case of piston three, the configuration of two ball valves did not fit the piston. A configuration with small balls would seem to be a better option. However, this number is still used, due to the lack of knowledge of the ball valve characteristics when they are scaled down.

The last limitation of this model is again concerning the actuator. Due to the actuator being a simple actuation mechanism, several physical phenomena are not, yet, modelled. For example, the friction in the pulley is not modelled. However, the friction does influence the efficiency and should be taken into account when the complete actuator is designed.

7.3 Recommendations

To be able to fully understand the complete multi piston pumping system of the Ocean Grazer, several follow up researches can be done. However, besides follow up research, one can also combine researches that are already done with this research.

As mentioned in the previous two paragraphs, the actuation mechanism is a simple version and is useful to get an insight in the different combinations. However, there are still physical phenomena that are currently not modelled. Therefore, a new research can be performed that investigates the actuation mechanism and models this part of the pumping system. By investigating the actuator, it might be possible to set the actuators to active with a specific wave input.

Furthermore, it can be concluded that the characteristics of the ball valves need more investigation. To be able to scale the ball valves in both the check valve and piston valve, this research is needed. Besides, to design the check valves and piston valves for the experimental setup an investigation into the ball valves might be needed.

Another factor in the total system that can be added to this research is the floater that is connected to the system. Investigating the role that the floater plays in the total system is important, especially when the total system is scaled to a larger size. When the floater is incorporated in the model, the complete power generation of the Ocean Grazer can be simulated. This recommendation was also made in the master thesis of R. Zaharia in case of the SPP model. However, this recommendation is also applicable on the MPP model.

As a last recommendation, it might be useful to combine this research with the research of K. Paapst and R. Folkertsma who investigated the experimental results of the experimental setup and the slamming phenomena respectively. When the project reaches the stage where the system is scaled up, the research of J. Jonker, who investigated the scaling of the SPP system, can be considered to investigate the scaling of the MPP system.

Bibliography

1. United Nations Framework Convention on Climate Change. Paris agreement. . 2015.

2. International Energy Agency. The world energy outlook 2017. . 2017.

3. Hussain A, Arif SM, Aslam M. Emerging renewable and sustainable energy technologies: State of the art. *Renewable and Sustainable Energy Reviews*. 2017;71:12-28.

4. Lehmann M, Karimpour F, Goudey CA, Jacobson PT, Alam M. Ocean wave energy in the united states: Current status and future perspectives. *Renewable & sustainable energy reviews*. 2017;74:1300-1313.

5. van de Weijer B. Delftse studenten winnen hyperloopwedstrijd van SpaceX. *De Volkskrant*. 30 January 2017.

6. Pecher A, Kofoed JP. *Handbook of ocean wave energy.* [Cham], Switzerland : Springer Open,; 2017:1 online resource (xix, 287 pages) : illustrations (some color).

7. U.S. Department of Energy - Office of Environmental Management. Technology readiness assessment (TRA) / technology maturation plan (TMP) process guide. . 2008.

8. van Rooij M. *Experimental validation of dynamical contact models of the ocean grazer.* [Master of Science]. Rijksuniversiteit Groningen; 2015.

9. The Ocean Grazer company. Ocean grazer website. http://www.oceangrazer.com/. Updated 2018. Accessed 04/15, 2018.

10. Vakis AI, Anagnostopoulos JS. Mechanical design and modeling of a single-piston pump for the novel power take-off system of a wave energy converter. *Renewable Energy*. 2016;96(Part A):531-547.

11. Wei Y, Barradas-Berglind JJ, van Rooij M, Prins WA, Jayawardhana B, Vakis AI. Investigating the adaptability of the multi-pump multi-piston power take-off system for a novel wave energy converter. *Renewable Energy*. 2017;111:598-610.

12. Jackson MC. *Systems thinking : Creative holism for managers.* Chichester, West Sussex ;: John Wiley & Sons,; 2003:xxiii, 352 pages : illustrations ; 24 cm.

13. Wieringa R, Heerkens H. Designing requirements engineering research. . 2007:36-48.

14. Zaharia RM. *Understanding of the single-piston pump inside the ocean grazer.* [Master of Science]. Groningen: Rijksuniversiteit Groningen; 2018.

15. Åström KJ, Murray RM. *Feedback systems : An introduction for scientists and engineers.* Princeton : Princeton University Press,; 2008:xii, 396 pages : illustrations, map ; 27 cm.

16. The MathWorks I. https://nl.mathworks.com/help/physmod/hydro/ref/checkvalve.html. Updated 2018. Accessed 05/23, 2018.

17. Nakayama Y, Boucher RF. *Introduction to fluid mechanics.* Oxford ;: Butterworth Heine-mann,; 1999:1 online resource (xii, 308 pages) : illustrations (some color).

Symbols and Abbreviations

Abbreviations

CV	–	*Check Valve*
MPP	–	*Multi Piston Pump*
PV	–	*Piston Valve*
SPP	–	*Single Piston Pump*
WEC	–	*Wave Energy Converter*

Symbols

A	–	*Area*
C_D	–	*Drag coefficient*
c	–	*Damping factor*
d	–	*Diameter*
F	–	*Force*
f_n	–	*Natural Frequency*
g	–	*Gravitational constant*
k	–	*Spring constant*
m	–	*Mass*
p	–	*Pressure*
Q	–	*Flow rate*
q	–	*Displacement*
$\dot{q}$	–	*Velocity*
Re	–	*Reynolds number*
r	–	*Radius*
TWh	–	*Terawatt hour*
t	–	*Time*
V	–	*Volume*
v	–	*Velocity*
λ	–	*Decay*
ω	–	*Frequency*
ω_0	–	*Radian frequency*
μ	–	*Viscosity*
ρ	–	*Density*
ζ	–	*Damping ratio*

Appendices

Appendix 1. *Matlab script: initialization_parameters_MPP_V2.m*

Appendix 2. *Matlab script: calculationsSimulink_MPP_V2.m*

Appendix 3. *Matlab script: run_simulinkmodel_MPP_V2.m*

Appendix 4. *Matlab script: plot_Simulinkresults_MPP_V2.m*

Appendix 5. *Simulink model: MPPmodel_V2*

Appendix 6. *Logbook*

Appendix 7. *Gantt chart of activities (Planning block II)*

Appendix 8. *Bill of Materials (BOM) and technical drawing*

Appendix 1. Matlab Script: initialization_parameters_MPP_v2.m

```
%----------------------------------------------------------------------
% INITIATE DATA - MPPmodel_V2 _ Simulink model
% Model V 2.0
% Created by R.M. Zaharia
% Adjusted by L.Y. Hut
%----------------------------------------------------------------------
% if exist('read_data.m','file')
%     close all
% else
%     clear all
%     clc
%     close all
% end
%     clear all
%     close all
%     clc
%----------------------------------------------------------------------
% Ask for the values of the water level
%----------------------------------------------------------------------
    prompt = {'Initial level lower reservoir [m] :', ...
        'Initial level uppper reservoir [m]:','Motor arm setting:', ...
        'Motor frequency','Number of strokes:'};
    dlg_title = 'Initializiations';
    num_lines = 1;
    defaultans = {'1.197','1.658','7','15','3'};
    answer = inputdlg(prompt,dlg_title,[1 50],defaultans);

%----------------------------------------------------------------------
% Save the values in their own variable name
%----------------------------------------------------------------------
    for i = 1:size(answer)
        data(i) = str2num(answer{i});
    end

    init_low = data(1);
    init_up = data(2);
    motor_set = data(3);
    motor_freq = data(4);
    strokes = data(5);


%----------------------------------------------------------------------
% Switches and changeable parameters
%----------------------------------------------------------------------
   cv1_amount = 12;                                % Number of ball valves
   pv1_amount = 6;

   cv2_amount = 7;
   pv2_amount = 4;

   cv3_amount = 4;
   pv3_amount = 2;

%----------------------------------------------------------------------
% Simulation Parameter
%----------------------------------------------------------------------

    rho = 998.159;                     % [kg/m^3] Density of working fluid
```

```
    g = 9.81;                         % [m/s^2] Gravitational constant

%--------------------------------------------------------------------
% Motor Parameters
%--------------------------------------------------------------------
    m_setting = motor_set;
    f_setting = motor_freq;
    % Velocity of the motor based on the settings
    omega = (2*pi*(0.15+0.05*m_setting)/100)*f_setting;
    % amplitude
    amp = ((2*pi*(0.15+0.05*m_setting)^2)/100)*f_setting;
    bias = 0;                                   % bias
    freq = omega;                               % [rad/sec] frequency
    phase = 0;                                  % [rad] Phase
    samp = 0;                                   % Sample time
    time = ((2*pi)/(abs(omega)))*strokes;       % [s] Running Time

%--------------------------------------------------------------------
% Reservoir Data
%--------------------------------------------------------------------
    u_reservoir_width = 1.56;                % [m] length of the reser-
voir
    u_reservoir_length = 1.97;               % [m] width of the reservoir
    u_reservoir_height = 1;                  % [m] height of the reser-
voir
    init_height_u = init_up - 1.478;

    upper_reservoir_volume = init_height_u*u_reservoir_width*u_reser-
voir_length;
                                             % [m^3] Volume of the lower
reservoir


    l_reservoir_width = 1.77;                % [m] width of the reservoir
    l_reservoir_length = 2.27;               % [m] length of the reser-
voir
    l_reservoir_height = 1;                  % [m] height of the reser-
voir
    init_height_l = init_low - 0.85;

    lower_reservoir_volume = init_height_l*l_reservoir_width*l_reser-
voir_length;
                                             % [m^3] Volume of the higher
reservoir


    tank_cross_section_higher = u_reservoir_width * u_reservoir_length;

    tank_cross_section_lower = l_reservoir_width * l_reservoir_length;

    reservoir_pipe_diameter = 0.18;           % [m] Diameter of pipe out
and in reservoir

%--------------------------------------------------------------------
% Pipe Data
%--------------------------------------------------------------------
    % In this model, there are currently three pipes. The first from the
lower
    % reservoir to the check valve. The second from the check valve to the
```

```
    % piston. The third from the piston to the higher reservoir.

    % Materials
    % https://www.engineeringtoolbox.com/surface-roughness-ventilation-
ducts-d_209.html
    % https://www.researchgate.net/publication/230737556_Analysis_of_Sur-
face_Roughness_for_Laser_Cutting_on_Acrylic_Sheets_using_Response_Sur-
face_Method
    % http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-
43662017000300143
    PVC = 0.0015e-3;                            % [m] Internal surface
roughness height
    metal = 0.015e-3;                           % [m] Internal surface
roughness height
    acryl = 5e-6;                               % [m] Internal surface
roughness height

    % Pipe (1) - material: PVC
    diameter_pipe_1 = 0.18;                     % [m] Diameter of the first
pipe
    height_A_1 = 0.8;                           % [m] height of inlet pipe
wrt reference plane
    height_B_1 = 0.244;                           % [m] height of inlet
pipe wrt reference plane
    pipe_length_1 = height_A_1 - height_B_1;    % [m] Lenght of the first
pipe


if piping_choice > 0
    % Bend (2) - material: PVC
    diameter_pipe_2 = 0.18;                     % [m] Diameter of bend
    area_pipe_2 = pi*(diameter_pipe_2/2)^2;
    bend_radius_2 = 0.3;                        % [m] Bend radius
    bend_angle_2 = 90;                          % [degrees] Bend angle

    % Pipe (3) - material: PVC
    diameter_pipe_3 = 0.18;
    pipe_length_3 = 1.41;
    height_A_3 = 0.244;
    height_B_3 = 0.244;

    % Butterfly Valve (4) - material: PVC
    length_pipe_4 = 0.2;
    diameter_valve_4 = 0.16;
    area_valve_4 = pi*(diameter_valve_4/2)^2;
    flow_discharge_4 = 0.856;

    % Pipe 1-4 are common. After Pipe 4 there are three paths. Pipes are
    % named correspondingly. F.E. the first pipe of the first path is pipe
    % nr. 1.1. Pipe 5 and 6 are an exception, these are common pipe for two
paths.
    % Path 1 corresponds with Piston 1, Path 2 with Piston 2 and Path 3
    % with Piston 3.

    % Path 1
    % Pipe (5) - material: PVC
    diameter_pipe_5 = 0.20;
    pipe_length_5 = 0.4213;
    height_A_5 = 0.244;
    height_B_5 = 0.244;
```

```
    % Pipe (6) - material: PVC
    diameter_pipe_6 = 0.19;
    pipe_length_6 = 0.26;
    height_A_6 = 0.244;
    height_B_6 = 0.244;

    % Pipe (P1.1) - Material: PVC
    diameter_pipe_p1_1 = 0.18;                          % [m] Diameter of
bend
    bend_radius_p1_1 = 0.18;                            % [m] Bend radius
    bend_angle_p1_1 = 90;                               % [degrees] Bend
angle
    area_pipe_p1_1 = pi*(diameter_pipe_p1_1/2)^2;       % [m^2] surface
area

    % Pipe (P1.2) - Material: PVC
    diameter_pipe_p1_2 = 0.19;
    pipe_length_pipe_p1_2 = 0.66;
    Height_A_p1_2 = 0.244;
    Height_B_p1_2 = 0.244;

    % Pipe (P1.3) - Material: PVC
    diameter_pipe_p1_3 = 0.1602;
    bend_radius_p1_3 = 0.2934;
    bend_angle_p1_3 = 90;
    area_pipe_p1_3 = pi*(diameter_pipe_p1_3/2)^2;

    % Pipe (P1.4) - Material: PVC
    diameter_pipe_p1_4 = 0.20;
    pipe_length_pipe_p1_4 = 0.14585;
    Height_A_p1_4 = 0.69315;
    Height_B_p1_4 = 0.839;

    % Path 2
    % Pipe (P2.1) - Material: PVC
    diameter_pipe_p2_1 = 0.18;
    bend_radius_p2_1 = 0.18;
    bend_angle_p2_1 = 90;
    area_pipe_p2_1 = pi*(diameter_pipe_p2_1/2)^2;

    % Pipe (P2.2) - Material: PVC
    diameter_pipe_p2_2 = 0.15;
    pipe_length_p2_2 = 0.26;
    height_A_p2_2 = 0.244;
    height_B_p2_2 = 0.244;

    % Pipe (P2.3) - Material: PVC
    diameter_pipe_p2_3 = 0.124;
    bend_radius_p2_3 = 0.31556;
    bend_angle_p2_3 = 15;
    area_pipe_p2_3 = pi*(diameter_pipe_p2_3/2)^2;

    % Pipe (P2.4) - Material: PVC
    diameter_pipe_p2_4 = 0.15;
    pipe_length_p2_4 = 0.40;
    height_A_p2_4 = 0.244;
    height_B_p2_4 = 0.244;
```

```
% Pipe (P2.5) - Material: PVC
diameter_pipe_p2_5 = 0.124;
bend_radius_p2_5 = 0.23030;
bend_angle_p2_5 = 90;
area_pipe_p2_5 = pi*(diameter_pipe_p2_5/2)^2;

% Pipe (P2.6) - Material: PVC
diameter_pipe_p2_6 = 0.15;
pipe_length_p2_6 = 0.26;
height_A_p2_6 = 0.50548;
height_B_p2_6 = 0.76548;

% Pipe (P2.7) - Material: PVC
diameter_pipe_p2_7 = 0.15;
pipe_length_p2_7 = 0.07352;
height_A_p2_7 = 0.76548;
height_B_p2_7 = 0.839;

% Path 3
% Pipe (P3.1) - Material: PVC
diameter_pipe_p3_1 = 0.20;
pipe_length_p3_1= 0.4213;
height_A_p3_1 = 0.244;
height_B_p3_1 = 0.244;

% Pipe (P3.2) - Material: PVC
diameter_pipe_p3_2 = 0.18;
bend_radius_p3_2 = 0.18;
bend_angle_p3_2 = 90;
area_pipe_p3_2 = pi*(diameter_pipe_p3_2/2)^2;

% Pipe (P3.3) - Material: PVC
diameter_pipe_p3_3 = 0.20;
lenght_pipe_p3_3 = 0.4213;
heigth_A_p3_3 = 0.244;
height_B_p3_3 = 0.244;

% Pipe (P3.4) - Material: PVC
diameter_pipe_p3_4 = 0.086;
bend_radius_p3_4 = 0.25271;
bend_angle_p3_4 = 15;
area_pipe_p3_4 = pi*(diameter_pipe_p3_4/2)^2;

% Pipe (P3.5) - Material: PVC
diameter_pipe_p3_5 = 0.10;
lenght_pipe_p3_5 = 0.35;
height_A_p3_5 = 0.244;
height_B_p3_5 = 0.244;

% Pipe (P3.6) - Material: PVC
diameter_pipe_p3_6 = 0.086;
bend_radius_p3_6 = 0.154;
bend_angle_p3_6 = 90;
area_pipe_p3_6 = pi*(diameter_pipe_p3_6/2)^2;

% Pipe (P3.7) - Material: PVC
diameter_pipe_p3_7 = 0.10;
length_pipe_p3_7 = 0.35;
height_A_p3_7 = 0.41878;
height_B_p3_7 = 0.76878;
```

```
     % Pipe (P3.8) - Material: PVC
    diameter_pipe_p3_8 = 0.10;
    length_pipe_p3_8 = 0.07022;
    height_A_p3_8 = 0.76878;
    height_B_p3_8 = 0.839;

    % Butterfly Valve (8) - material: PVC
    diameter_valve_8 = 0.16;
    flow_discharge_8 = 0.856;

else
% ------------ SMALL PIPING PARAMETERS ARE FOUND HERE!!!! -----------------
    % Bend (2) - material: PVC
    diameter_pipe_2 = 0.10;                      % [m] Diameter of bend
    area_pipe_2 = pi*(diameter_pipe_2/2)^2;
    bend_radius_2 = 0.1;                         % [m] Bend radius
    bend_angle_2 = 90;                           % [degrees] Bend angle

    % Pipe (3) - material: PVC
    diameter_pipe_3 = 0.10;
    pipe_length_3 = 1.41;
    height_A_3 = 0.2;
    height_B_3 = 0.2;

    % Butterfly Valve (4) - material: PVC
    length_pipe_4 = 0.2;
    diameter_valve_4 = 0.10;
    area_valve_4 = pi*(diameter_valve_4/2)^2;
    flow_discharge_4 = 0.856;

    % Pipe (5) - material: PVC
    diameter_pipe_5 = 0.10;
    pipe_length_5 = 0.94;
    height_A_5 = 0.2;
    height_B_5 = 0.2;

    % Bend (6) - material: PVC
    diameter_pipe_6 = 0.10;                      % [m] Diameter of bend
    bend_radius_6 = 0.18;                        % [m] Bend radius
    bend_angle_6 = 90;                           % [degrees] Bend angle
    area_pipe_6 = pi*(diameter_pipe_6/2)^2;      % [m^2] surface area

    % Pipe (7) - material: PVC
    diameter_pipe_7 = 0.10;
    pipe_length_7 = 0.175;
    height_A_7 = 0.2;
    height_B_7 = 0.2;

    % Butterfly Valve (8) - material: PVC
    diameter_valve_8 = 0.16;
    flow_discharge_8 = 0.856;

    % Pipe (9) - material: PVC
    diameter_pipe_9 = 0.10;
    pipe_length_9 = 0.755;
    height_A_9 = 0.2;
    height_B_9 = 0.2;
```

```
    % Bend (10) - material: PVC
    diameter_pipe_10 = 0.10;                    % [m] Diameter of bend
    bend_radius_10 = 0.1;                       % [m] Bend radius
    bend_angle_10 = 90;                         % [degrees] Bend angle
    area_pipe_10 = pi*(diameter_pipe_10/2)^2;   % [m^2] surface area
end
% ------------- END SMALL PIPING PARAMETERS ------------------------------
    % Pipe (11) - material: PVC
    diameter_pipe_11 = 0.18;
    height_A_11 = 0.2;
    height_B_11 = 0.86;
    pipe_length_11 = height_B_11 - height_A_11;

    % Piping from Check valve to the Piston Pipe
    % CV-PP 1 - Material: PVC
    diameter_pipe_CV_PP_1 = 0.19;
    length_pipe_CV_PP_1 = 0.954;
    height_CV_PP_1_A = 1.505;
    height_CV_PP_1_B = 2.459;

    % CV-PP 2 - Material: PVC
    diameter_pipe_CV_PP_2 = 0.15;
    length_pipe_CV_PP_2 = 0.954;
    height_CV_PP_2_A = 1.505;
    height_CV_PP_2_B = 2.459;

    % CV-PP 3 - Material: PVC
    diameter_pipe_CV_PP_3 = 0.10;
    length_pipe_CV_PP_3 = 0.954;
    height_CV_PP_3_A = 1.505;
    height_CV_PP_3_B = 2.459;

    diameter_pipe_14 = 0.18;
    pipe_length_14 = 0.01;
    height_A_14 = 4.059;
    height_B_14 = 4.069;


    % Pipe (14) - material: Metal - Piston Cylinder
    % See other section

%-------------------------------------------------------------------
% Check Valve Data
%-------------------------------------------------------------------

% Check valve 1 (Check Valve Corresponding to Piston 1)

    radius_cv1 = 0.015;                         % [m] Radius passage area
Check Valve
    max_passage_area_cv1 = pi*(radius_cv1)^2;   % [m^2] Maximum passage
area Check Valve
    crack_cv1 = 1e1;                            % [Pa] Cracking Pressure
CV
    max_cv1 = 1e4;                              % [Pa] Maximum opening
pressure CV
    leak_cv1 = 1e-10;                           % [m^2] Leakage area
    opening_cv1 = 0.1;                          % [s] Opening time con-
stant
    init_area_cv1 = 1e-10;                      % [m^2] initial area CV
```

```
    % Piping before and after the check valve
    lenght_cv1_pipe_down = 0.318;                % [m] down pipe cv length
including sealing ring
    cv1_down_A = 0.839;
    cv1_down_B = 1.157;
    radius_cv1_pipe_down = 0.192;                % [m] radius down cv pipe

    lenght_cv1_pipe_up = 0.318;                  % [m] upper cv pipe
length including sealing ring
    cv1_up_A = 1.187;
    cv1_up_B = 1.505;
    radius_cv1_pipe_up = 0.192;                  % [m] radius upper cv
pipe


    % Fluid inertia should also be taken into account for the CV (see block
    % there).
    height_cv1 = 0.030;                          % [m] height of the CV


% Check valve 2 (Check Valve Corresponding to Piston 2)
    radius_cv2 = 0.015;                          % [m] Radius passage area
Check Valve
    max_passage_area_cv2 = pi*(radius_cv2)^2;    % [m^2] Maximum passage
area Check Valve
    crack_cv2 = 1e1;                             % [Pa] Cracking Pressure
CV
    max_cv2 = 1e4;                               % [Pa] Maximum opening
pressure CV
    leak_cv2 = 1e-10;                            % [m^2] Leakage area
    opening_cv2 = 0.1;                           % [s] Opening time con-
stant
    init_area_cv2 = 1e-10;                       % [m^2] initial area CV

    % Piping before and after the check valve
    length_cv2_pipe_down = 0.318;                % [m] down pipe cv length
including sealing ring
    cv2_down_A = 0.839;
    cv2_down_B = 1.157;
    radius_cv2_pipe_down = 0.142;                % [m] diameter down cv
pipe

    length_cv2_pipe_up = 0.318;                  % [m] upper pipe cv
length including sealing ring
    cv2_up_A = 1.187;
    cv2_up_B = 1.505;
    radius_cv2_pipe_up = 0.142;                  % [m] diameter upper cv
pipe


    % Fluid inertia should also be taken into account for the CV (see block
    % there).
    height_cv2 = 0.030;                           % [m] height of the CV


% Check valve 3 (Check Valve Corresponding to Piston 3)
    radius_cv3 = 0.015;                          % [m] Radius passage area
Check Valve
```

```
    max_passage_area_cv3 = pi*(radius_cv3)^2;      % [m^2] Maximum passage
area Check Valve
    crack_cv3 = 1e1;                                % [Pa] Cracking Pressure
CV
    max_cv3 = 1e4;                                  % [Pa] Maximum opening
pressure CV
    leak_cv3 = 1e-10;                               % [m^2] Leakage area
    opening_cv3 = 0.1;                              % [s] Opening time con-
stant
    init_area_cv3 = 1e-10;                          % [m^2] initial area CV

    % Piping before and after the check valve
    length_cv3_pipe_down = 0.318;                   % [m] down pipe cv length
including sealing ring
    cv3_down_A = 0.839;
    cv3_down_B = 1.157;
    radius_cv3_pipe_down = 0.100;                   % [m] diameter down cv
pipe

    length_cv3_pipe_up = 0.318;                     % [m] upper pipe cv
length including sealing ring
    cv3_up_A = 1.187;
    cv3_up_B = 1.505;
    radius_cv3_pipe_up = 0.100;                     % [m] diameter upper cv
pipe


    % Fluid inertia should also be taken into account for the CV (see block
    % there).
    height_cv3 = 0.030;                             % [m] height of the CV

%-----------------------------------------------------------------------
% Piston Pump Parameters (without Valve)
%-----------------------------------------------------------------------

    % Piston Pump 1
    piston_1_mass = 25;                             % [kg] Mass piston/rod
    radius_rod_1 = 0.0025;                          % [m] Radius rod
    Est_rod_1 = 210e9;                              % [Pa] Young's Modulus of
steel
    length_rod_1 = 5;                               % [m] Length rod
    zeta_1 = 25;                                    % Rod damping ratio
    K1_1 = (pi*(radius_rod_1^2)*Est_rod_1)/length_rod_1;
    C1_1 = 2*zeta_1*sqrt(piston_1_mass*K1_1);
%     K2 = rho*g*(pi*(0.09)^2);

    %l_pipe_t = pipe_length_1 + pipe_length_3 + pipe_length_5 + ...
     %   pipe_length_7 + pipe_length_9 + pipe_length_11 + pipe_length_12
...
      %  + pipe_length_13 + 4;
    %m_pipes = pi * (0.09^2) * l_pipe_t * rho;

    %n_freq = sqrt((2*g)/l_pipe_t);
    %c_ratio = 0.098;
    % Damping water column
    %C1_2 = 2*0.5*m_pipes*n_freq*c_ratio;

    % Stiffness
    %K1_2 = 2*rho*(pi*0.095^2)*g;
```

```
%     damper = (C1_1 + C1_2)/300;
%     spring = (1/((1/K1_1)+(1/K1_2)))*300  ;

    piston_1_rad = 0.09485;                          % [m] Radius of piston
    piston_1_area = pi*piston_1_rad^2;               % [m^2] Piston area
    length_piston_1_pipe = 1.6;                      % length piston cylinder
    diameter_piston_1_pipe = 0.19;
    piston_1_pipe_A = 2.459;
    piston_1_pipe_B = 4.059;


    % Piston Pump 2
    piston_2_mass = 15.4;                             % [kg] Mass piston/rod
    radius_rod_2 = 0.0025;                          % [m] Radius rod
    Est_rod_2 = 210e9;                              % Young's Modulus of steel
    length_rod_2 = 5;                               % [m] Length rod
    zeta_2 = 25;                                    % Rod damping ratio
    K2_1 = (pi*(radius_rod_2^2)*Est_rod_2)/length_rod_2;
    C2_1 = 2*zeta_2*sqrt(piston_2_mass*K2_1);
%     K2 = rho*g*(pi*(0.09)^2);

    %l_pipe_t = pipe_length_1 + pipe_length_3 + pipe_length_5 + ...
    %    pipe_length_7 + pipe_length_9 + pipe_length_11 + pipe_length_12
...
    %    + pipe_length_13 + 4;
    %m_pipes = pi * (0.09^2) * l_pipe_t * rho;

    %n_freq = sqrt((2*g)/l_pipe_t);
    %c_ratio = 0.098;
    % Damping water column
    %C2_2 = 2*0.5*m_pipes*n_freq*c_ratio;

    % Stiffness
    %K2_2 = 2*rho*(pi*0.095^2)*g;

%     damper = (C2_1 + C2_2)/300;
%     spring = (1/((1/K2_1)+(1/K2_2)))*300  ;

    piston_2_rad = 0.06985;                         % [m] Radius of piston
    piston_2_area = pi*piston_2_rad^2;              % [m^2] Piston area
    length_piston_2_pipe = 1.6;                     % length piston cylinder
    diameter_piston_2_pipe = 0.14;
    piston_2_pipe_A = 2.459;
    piston_2_pipe_B = 4.059;


    % Piston Pump 3
    piston_3_mass = 7.8;                             % [kg] Mass piston/rod
    radius_rod_3 = 0.0025;                          % [m] Radius rod
    Est_rod_3 = 210e9;                              % Young's Modulus of steel
    length_rod_3 = 5;                               % [m] Length rod
    zeta_3 = 25;                                    % Rod damping ratio
    K3_1 = (pi*(radius_rod_3^2)*Est_rod_3)/length_rod_3;
    C3_1 = 2*zeta_3*sqrt(piston_3_mass*K3_1);
%     K2 = rho*g*(pi*(0.09)^2);

    %l_pipe_t = pipe_length_1 + pipe_length_3 + pipe_length_5 + ...
```

```
    %    pipe_length_7 + pipe_length_9 + pipe_length_11 + pipe_length_12
...
    %    + pipe_length_13 + 4;
    %m_pipes = pi * (0.09^2) * l_pipe_t * rho;

    %n_freq = sqrt((2*g)/l_pipe_t);
    %c_ratio = 0.098;
    % Damping water column
    %C3_2 = 2*0.5*m_pipes*n_freq*c_ratio;

    % Stiffness
    %K3_2 = 2*rho*(pi*0.095^2)*g;

%      damper = (C3_1 + C3_2)/300;
%      spring = (1/((1/K3_1)+(1/K3_2)))*300  ;

    piston_3_rad = 0.04985;                      % [m] Radius of piston
    piston_3_area = pi*piston_3_rad^2;           % [m^2] Piston area
    length_piston_3_pipe = 1.6;                  % length piston cylinder
    diameter_piston_3_pipe = 0.10;
    piston_3_pipe_A = 2.459;
    piston_3_pipe_B = 4.059;

%---------------------------------------------------------------------
% Valve in Piston Parameters (pv = piston valve)
%---------------------------------------------------------------------

    % Piston Valve 1
%     total_area_pv = pi*0.05^2;
%     max_passage_area_pv = total_area_pv/6;
    radius_pv1 = 0.015;                            % [m] Radius passage
area Check Valve
    max_passage_area_pv1 = pi*(radius_pv1)^2;      % [m^2] Maximum passage
area Check Valve
    crack_pv1 = 0.1e1;                             % [Pa] Cracking Pressure
CV
    max_pv1 = 2.2e4;                               % [Pa] Maximum opening
pressure CV
    leak_pv1 = 1e-10;                              % [m^2] Leakage area
    opening_pv1 = 0.1;                             % [s] Opening time con-
stant
    init_area_pv1 = 1e-10;                         % [m^2] initial area CV

    piston_1_height = 0.1395;

     % Vertical piping section 1
    length_section_1_p1 = pipe_length_1 + bend_radius_2;
    % Vertical piping section 2
    length_section_2_p1 = bend_radius_p1_3 + pipe_length_pipe_p1_4 +
length_cv3_pipe_down + ...
        lenght_cv1_pipe_up + height_cv1 + length_pipe_CV_PP_1 + length_pis-
ton_1_pipe + ...
        pipe_length_14;

    % Piston Valve 2
 %     total_area_pv = pi*0.05^2;
 %     max_passage_area_pv = total_area_pv/6;
    radius_pv2 = 0.015;                            % [m] Radius passage
area Check Valve
```

```
    max_passage_area_pv2 = pi*(radius_pv2)^2;       % [m^2] Maximum passage
area Check Valve
    crack_pv2 = 0.1e1;                              % [Pa] Cracking Pressure
CV
    max_pv2 = 2.2e4;                                % [Pa] Maximum opening
pressure CV
    leak_pv2 = 1e-10;                               % [m^2] Leakage area
    opening_pv2 = 0.1;                              % [s] Opening time con-
stant
    init_area_pv2 = 1e-10;                          % [m^2] initial area CV

    piston_2_height = 0.1395;

     % Vertical piping section 1
    length_section_1_p2 = pipe_length_1 + bend_radius_2;
    % Vertical piping section 2
    length_section_2_p2 = bend_radius_p2_5 + pipe_length_p2_6 +
pipe_length_p2_7 + length_cv2_pipe_down + ...
        length_cv2_pipe_up + height_cv2 + length_pipe_CV_PP_2 + length_pis-
ton_2_pipe + ...
        pipe_length_14;

     % Piston Valve 3
 %     total_area_pv = pi*0.05^2;
 %     max_passage_area_pv = total_area_pv/6;
    radius_pv3 = 0.015;                             % [m] Radius passage
area Check Valve
    max_passage_area_pv3 = pi*(radius_pv3)^2;       % [m^2] Maximum passage
area Check Valve
    crack_pv3 = 0.1e1;                              % [Pa] Cracking Pressure
CV
    max_pv3 = 2.2e4;                                % [Pa] Maximum opening
pressure CV
    leak_pv3 = 1e-10;                               % [m^2] Leakage area
    opening_pv3 = 0.1;                              % [s] Opening time con-
stant
    init_area_pv3 = 1e-10;                          % [m^2] initial area CV

    piston_3_height = 0.1395;

     % Vertical piping section 1
    length_section_1_p3 = pipe_length_1 + bend_radius_2;
    % Vertical piping section 2
    length_section_2_p3 = bend_radius_p3_6 + length_pipe_p3_7 +
length_pipe_p3_8  + length_cv3_pipe_down + ...
        length_cv3_pipe_up + height_cv3 + length_pipe_CV_PP_3 + length_pis-
ton_3_pipe + ...
        pipe_length_14;

%-------------------------------------------------------------------
% Fluid Properties
%-------------------------------------------------------------------
    temp = 18;                                     % [deg Celcius] Tempera-
ture
    % Density, viscosity and bulk modulus depends on settings, but with 18
    % degrees it is as follows:
    % density (kg/m^3) = 998.159
    % Viscosity (cST): 1.05678
    % Bulk Modulus (Pa): 2.16651e+09
```

Appendix 2. Matlab script: calculationsSimulink_MPP_V2.m

```
%-------------------------------------------------------------------------
-----
% Plot results
% Created by R.M. Zaharia
% Adjusted by L.Y. Hut
%-------------------------------------------------------------------------
-----

% Switches
plot_force              = 1;
plot_Ppump              = 0;
plot_pressures          = 0;
plot_flowCV             = 1;
plot_flowPV             = 0;
plot_PVandCV            = 0;
plot_pressureCVPV       = 1;
plot_PVA                = 0;
plot_FVP                = 0;
plot_FVP_1              = 0;
plot_FA                 = 0;

% Determine the peaks of the position of the piston (for upstroke and
% downstroke). http://billauer.co.il/peakdet.html

maxamp = peakdet(position_motor,0.1,tout);
minamp = peakdet(-position_motor,0.1,tout);

if plot_force > 0
    figure
    plot(tout,motor_force1)
    hold on
    plot(tout,motor_force2)
    hold on
    plot(tout,motor_force3)
    title(['Piston Force (hydraulic head: ' num2str(delta_h1(1)) ' m)'])
    ylabel('Force [N]')
    xlabel('time (s)')
%     line([maxamp(1,1),maxamp(1,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
%     line([minamp(2,1),minamp(2,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
%     line([maxamp(2,1),maxamp(2,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
%     line([minamp(3,1),minamp(3,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
end



% plot pumping power
if plot_Ppump > 0
    figure
    plot(tout,Ppump_sim)
    title('Potential Power')
    ylabel('Pp(W)')
    xlabel('time (s)')
end
```

```
% plot pressures in reservoir
if plot_pressures > 0
    figure
    subplot(2,1,1)
    plot(tout,p1)
    title('Pressure (P1) Upper Reservoir')
    xlabel('time (s)')
    ylabel('Pressure [Pa]')

    subplot(2,1,2)
    plot(tout,p4)
    title('Pressure (P4) Lower Reservoir')
    xlabel('time (s)')
    ylabel('Pressure [Pa]')
end

if plot_flowCV > 0
    figure
    plot(tout,flow_cv1_total)
    hold on
    plot(tout, flow_cv2_total)
    hold on
    plot(tout, flow_cv3_total)
    title('Flow [m^3] through the check valve')
    xlabel('time (s)')
    ylabel('m^3')
    line([maxamp(1,1),maxamp(1,1)],[max(flow_cv1_total),min(flow_cv1_to-
tal)],'Color','r','LineStyle','--')
    line([minamp(2,1),minamp(2,1)],[max(flow_cv1_total),min(flow_cv1_to-
tal)],'Color','r','LineStyle','--')
    line([maxamp(2,1),maxamp(2,1)],[max(flow_cv1_total),min(flow_cv1_to-
tal)],'Color','r','LineStyle','--')
    line([minamp(3,1),minamp(3,1)],[max(flow_cv1_total),min(flow_cv1_to-
tal)],'Color','r','LineStyle','--')
end

if plot_flowPV > 0
    figure
    plot(tout,flow_pv1_total)
    hold on
    plot(tout, flow_pv2_total)
    hold on
    plot(tout, flow_pv3_total)
    title('Flow [m^3] through the piston valve')
    xlabel('time (s)')
    ylabel('m^3')
    line([maxamp(1,1),maxamp(1,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','r','LineStyle','--')
    line([minamp(2,1),minamp(2,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','r','LineStyle','--')
    line([maxamp(2,1),maxamp(2,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','r','LineStyle','--')
    line([minamp(3,1),minamp(3,1)],[max(flow_pv_total),min(flow_pv_to-
tal)],'Color','r','LineStyle','--')
end

if plot_PVandCV > 0
    figure
    plot(tout,flow_pv1_total,'r',tout,flow_cv1_total,'b')
    hold on
    plot(tout,flow_pv2_total,'r',tout,flow_cv2_total,'b')
```

```
    hold on
    plot(tout,flow_pv3_total,'r',tout,flow_cv3_total,'b')
    title('Flow throught the piston and check valve')
    xlabel('time(s)')
    ylabel('m^3')
    legend('Piston valve', 'Check valve')
    line([maxamp(1,1),maxamp(1,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','black','LineStyle','--')
    line([minamp(2,1),minamp(2,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','black','LineStyle','--')
    line([maxamp(2,1),maxamp(2,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','black','LineStyle','--')
    line([minamp(3,1),minamp(3,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','black','LineStyle','--')
end

if plot_PVA > 0
    figure
    subplot(3,1,1)
    plot(tout,piston_1_velocity)
    hold on
    plot(tout,piston_2_velocity)
    hold on
    plot(tout,piston_3_velocity)
    xlabel('time (s)')
    ylabel('velocity (m/s)')

    subplot(3,1,2)
    plot(tout,piston_1_pos)
    hold on
    plot(tout,piston_2_pos)
    hold on
    plot(tout,piston_3_pos)
    xlabel('time (s)')
    ylabel('Postion (m)')

    subplot(3,1,3)
    plot(tout,acceleration_motor)
    xlabel('time (s)')
    ylabel('Acceleration (m/s^2)')

end

if plot_pressureCVPV > 0
    figure
    subplot(2,1,1)
    plot(tout,p_cv1)
    hold on
    plot(tout,p_cv2)
    hold on
    plot(tout,p_cv3)
    title('Pressure difference over the check valve')
    xlabel('time (s)')
    ylabel('\Delta P (Pa)')
    vline(maxamp(1,1),'r','Start Upstroke');
    vline(minamp(2,1),'r','End Upstroke');



    subplot(2,1,2)
```

```
    plot(tout,p_pv1)
    hold on
    plot(tout,p_pv2)
    hold on
    plot(tout,p_pv3)
    title('Pressure difference over the piston valve')
    xlabel('time (s)')
    ylabel('\Delta P (Pa)')

    line([maxamp(1,1),maxamp(1,1)],[max(p_pv1),min(p_pv1)],'Col-
or','r','LineStyle','--')
    line([minamp(2,1),minamp(2,1)],[max(p_pv1),min(p_pv1)],'Col-
or','r','LineStyle','--')
    line([maxamp(2,1),maxamp(2,1)],[max(p_pv1),min(p_pv1)],'Col-
or','r','LineStyle','--')
    line([minamp(3,1),minamp(3,1)],[max(p_pv1),min(p_pv1)],'Col-
or','r','LineStyle','--')
end

if plot_FVP > 0
    figure
    subplot(3,1,1)
    plot(tout,(piston_pos1+abs(min(piston_pos1))))
    hold on
    plot(tout,(piston_pos2+abs(min(piston_pos1))))
    hold on
    plot(tout, (piston_pos3+abs(min(piston_pos3))))
    title('Velocity piston')
    xlabel('time (s)')
    ylabel('Postion (m)')
    line([maxamp(1,1),maxamp(1,1)],[max(piston_pos1+abs(min(pis-
ton_pos1))),min(piston_pos1+abs(min(piston_pos1)))],'Color','r','Lin-
eStyle','--')
    line([minamp(2,1),minamp(2,1)],[max(piston_pos1+abs(min(pis-
ton_pos1))),min(piston_pos1+abs(min(piston_pos1)))],'Color','r','Lin-
eStyle','--')
    line([maxamp(2,1),maxamp(2,1)],[max(piston_pos1+abs(min(pis-
ton_pos1))),min(piston_pos1+abs(min(piston_pos1)))],'Color','r','Lin-
eStyle','--')
    line([minamp(3,1),minamp(3,1)],[max(piston_pos1+abs(min(pis-
ton_pos1))),min(piston_pos1+abs(min(piston_pos1)))],'Color','r','Lin-
eStyle','--')

    subplot(3,1,2)
    plot(tout,motor_force1)
    hold on
    plot(tout,motor_force2)
    hold on
    plot(tout,motor_force3)

    title('Piston Force')
    ylabel('Force [N]')
    xlabel('time (s)')
    line([maxamp(1,1),maxamp(1,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
    line([minamp(2,1),minamp(2,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
    line([maxamp(2,1),maxamp(2,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
    line([minamp(3,1),minamp(3,1)],[max(motor_force1),min(mo-
tor_force1)],'Color','r','LineStyle','--')
```

```
    subplot(3,1,3)
    plot(tout,flow_pv1_total,'black',tout,flow_cv1_total,'b')
    hold on
    plot(tout,flow_pv2_total,'black',tout,flow_cv2_total,'b')
    hold on
    plot(tout,flow_pv3_total,'black',tout,flow_cv3_total,'b')
    title('Flow throught the piston and check valve')
    xlabel('time(s)')
    ylabel('m^3')
    legend('Piston valve', 'Check valve')
    line([maxamp(1,1),maxamp(1,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','r','LineStyle','--')
    line([minamp(2,1),minamp(2,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','r','LineStyle','--')
    line([maxamp(2,1),maxamp(2,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','r','LineStyle','--')
    line([minamp(3,1),minamp(3,1)],[max(flow_pv1_total),min(flow_pv1_to-
tal)],'Color','r','LineStyle','--')
end

if plot_FVP_1 > 0

    figure
    subplot(3,1,1)
    plot(tout,piston_1_velocity,'b')
    hold on
    plot(tout,piston_2_velocity,'b')
    hold on
    plot(tout,piston_3_velocity,'b')
    axis([0 30 -0.3 0.3])
    title('Velocity piston')
    xlabel('time (s)')
    ylabel('Postion (m)')
    vline(maxamp(1,1),'r','Start Downstroke');
    vline(minamp(2,1),'m','Switch D -> U');
    vline(maxamp(2,1),'black','End Upstroke')

    subplot(3,1,2)
    plot(tout,motor_force1)
    hold on
    plot(tout,motor_force2)
    hold on
    plot(tout,motor_force3)

    axis([0 30 -200 1200])
    title('Piston Force')
    ylabel('Force [N]')
    xlabel('time (s)')
    vline(maxamp(1,1),'r','Start Downstroke');
    vline(minamp(2,1),'m','Switch D -> U');
    vline(maxamp(2,1),'black','End Upstroke')

    subplot(3,1,3)
    plot(tout,flow_pv1_total,'black',tout,flow_cv1_total,'b')
    hold on
    plot(tout,flow_pv2_total,'black',tout,flow_cv2_total,'b')
    hold on
    plot(tout,flow_pv3_total,'black',tout,flow_cv3_total,'b')
    axis([0 30 -0.005 0.008])
```

```
    title('Flow throught the piston and check valve')
    xlabel('time(s)')
    ylabel('Flow m^3')
    legend('Piston valve', 'Check valve')
    vline(maxamp(1,1),'r','Start Downstroke');
    vline(minamp(2,1),'m','Switch D -> U');
    vline(maxamp(2,1),'black','End Upstroke')

%      set(gcf,'PaperUnits','centimeters');
%      set(gcf,'PaperSize',[20 20]);
%      set(gcf,'PaperPosition',[0 0 20 20]);
%
%      dir_pdf1='fig_pfv.pdf';
%      print(gcf,'-dpdf',dir_pdf1)
end

if plot_FA > 0
    figure
    subplot(2,1,1)
    plot(tout,acceleration_motor)
    title('Acceleration Piston')
    ylabel('Acceleration (m/s^2)')
    xlabel('time (s)')
    axis([0 30 -0.2 0.2])
    vline(maxamp(1,1),'r','Start Downstroke');
    vline(minamp(2,1),'m','Switch D -> U');
    vline(maxamp(2,1),'black','End Upstroke')

    subplot(2,1,2)
    plot(tout,motor_force1)
    hold on
    plot(tout,motor_force2)
    hold on
    plot(tout,motor_force3)
    title('Piston Force')
    ylabel('Force [N]')
    xlabel('time (s)')
    axis([0 30 -300 1200])
    vline(maxamp(1,1),'r','Start Downstroke');
    vline(minamp(2,1),'m','Switch D -> U');
    vline(maxamp(2,1),'black','End Upstroke')
end
```

Appendix 3. Matlab script: run_Simulinkmodel_MPP_V2.m

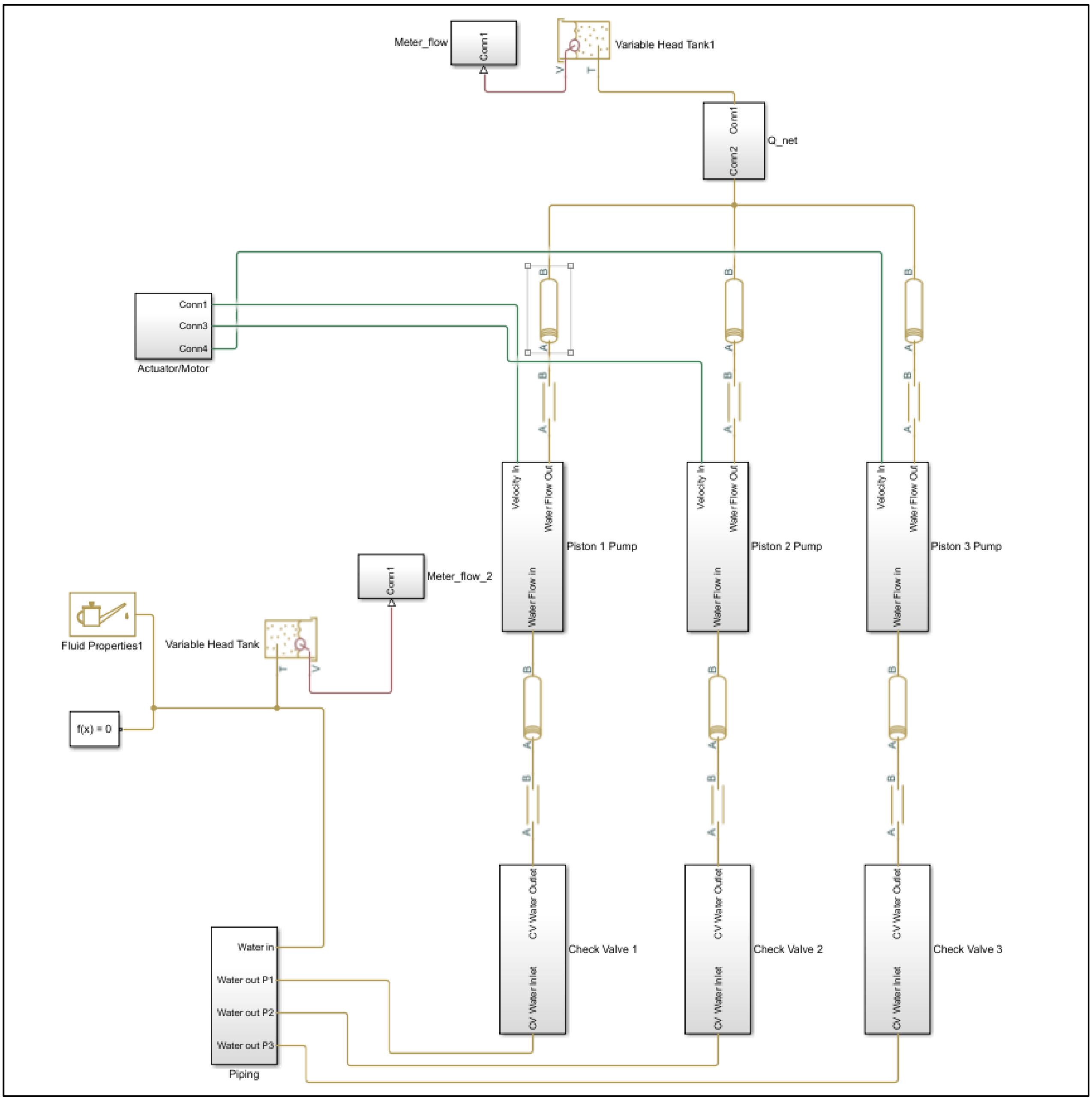
```
%----------------------------------------------------------------------
% Run model - MPPmodel_V2 _ Simulink model
% Model V 2.0
% Created by R.M. Zaharia
% Adjusted by L.Y. Hut
%----------------------------------------------------------------------
    clear all
    clc
    close all

% Switches
    calculations_data    = 1;    % Run the calculations
    display_data         = 1;    % Display the data
    plot_results         = 1;    % Plot results (see plot file)
    piping_choice        = 1;    % 1 is large piping, 0 is small piping

%----------------------------------------------------------------------
% Run the simulation
%----------------------------------------------------------------------
initialization_parameters_MPP_V2
    combination=[0 0 1];

    damper_coupling = 43000;                    % [N/(m/s)] Damper
    spring_coupling = 1;                        % [N/m] Spring

    damper1 = 43000;                            % [N/(m/s)] Damper
    spring1 = 1;                                % [N/m] Spring

    damper2 = 43000;                            % [N/(m/s)] Damper
    spring2 = 1;                                % [N/m] Spring

    damper3 = 43000;                            % [N/(m/s)] Damper
    spring3 = 1;                                % [N/m] Spring

    % Run the simulink model
    disp('Simulation is running...');
    options = simset('SrcWorkspace','current');
    sim('MPPmodel_V2',[],options);

%----------------------------------------------------------------------
% Determine the switches
%----------------------------------------------------------------------
if calculations_data > 0
    if display_data > 0
        disp_data = 1;
    else
        disp_data = 0;
    end
calculationsSimulink_MPP_V2
end

if plot_results > 0
    plot_simulinkresults_MPP_V2
end
```

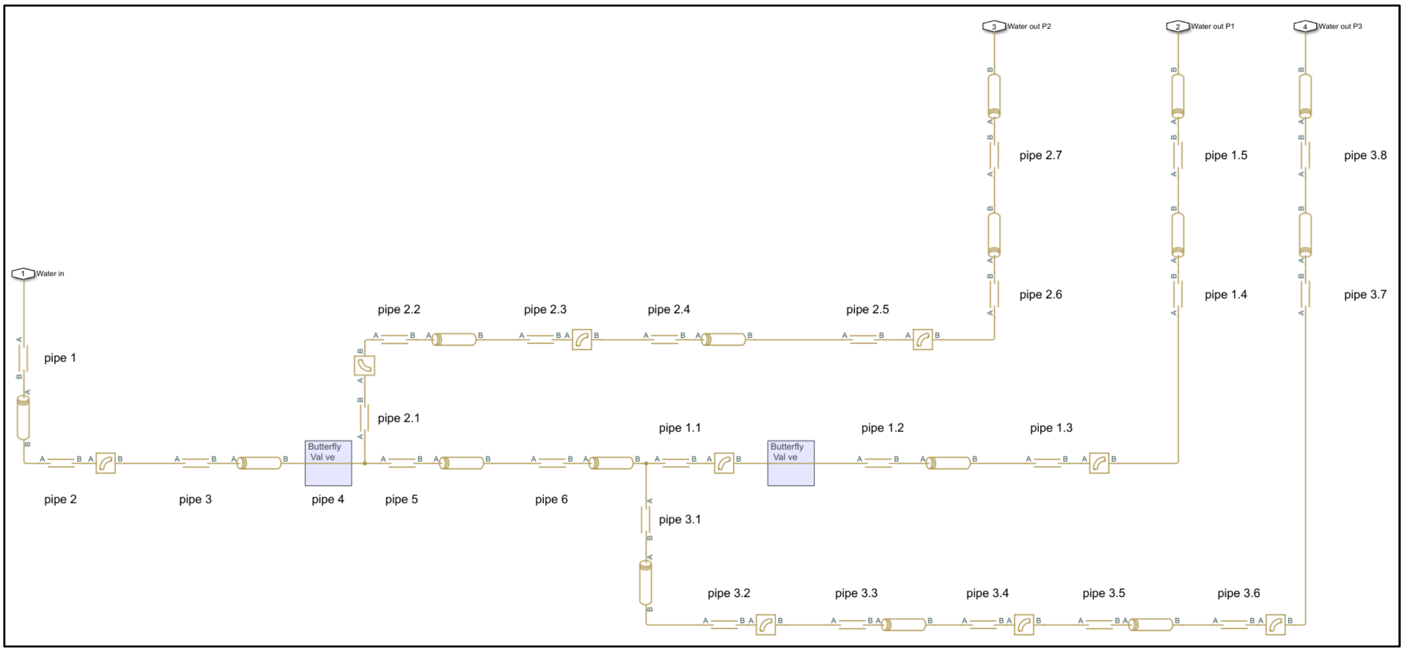
Appendix 4. Matlab script: plot_Simulinkresults_MPP_V2.m

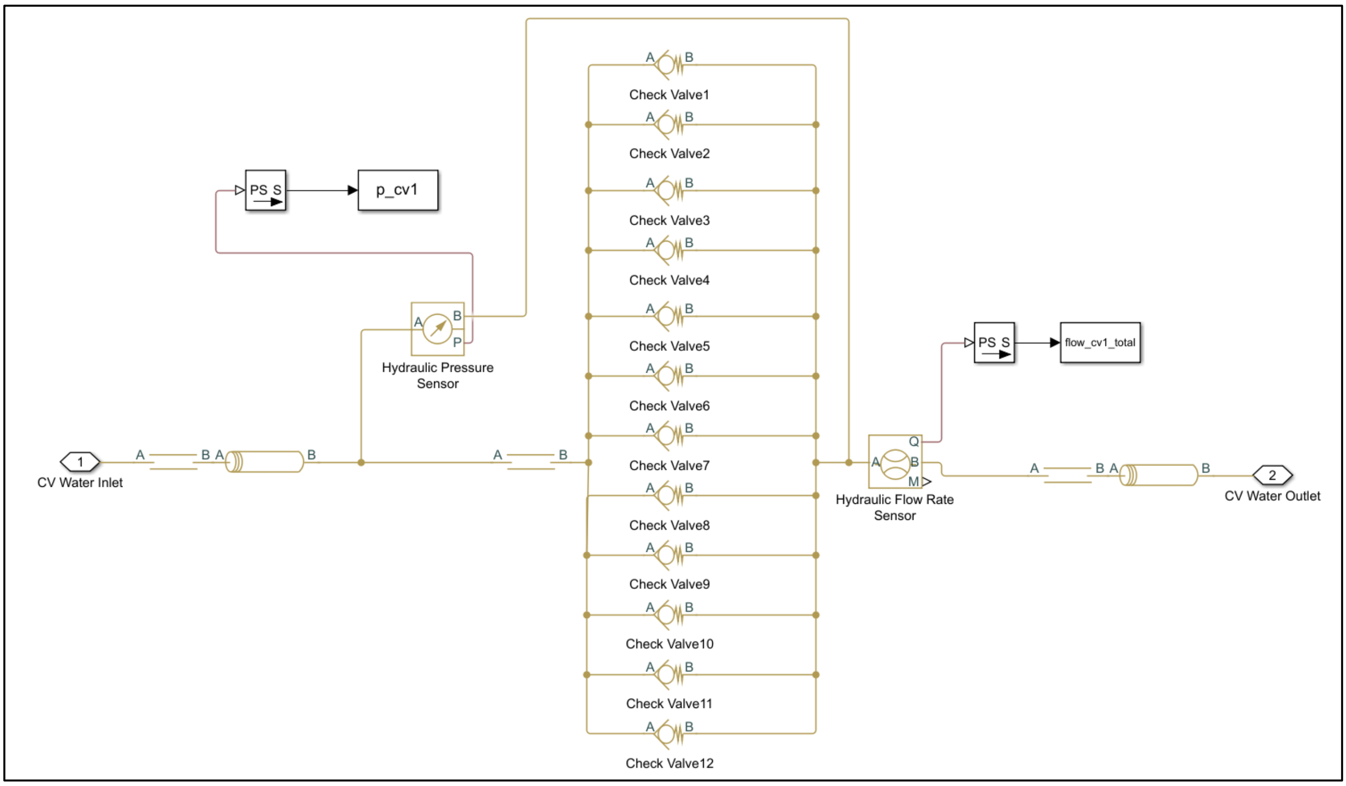
```
%-----------------------------------------------------------------------
% Calculations
% Created by R.M. Zaharia
% Adjusted by L.Y. Hut
%-----------------------------------------------------------------------

% Hydraulic head [m] - difference between both bottom reservoirs

for k = 1:size(tout)
    delta_h1(k) = l_upper_reservoir(k) + length_section_2_p1 - l_lower_res-
ervoir(k) - ...
        length_section_1_p1;
    delta_h2(k) = l_upper_reservoir(k) + length_section_2_p2 - l_lower_res-
ervoir(k) - ...
        length_section_1_p2;
    delta_h3(k) = l_upper_reservoir(k) + length_section_2_p3 - l_lower_res-
ervoir(k) - ...
        length_section_1_p3;
end


%Take only the force in the upstroke
    forceUp = zeros(size(tout));
    for i = 1:size(tout),
        if motor_force1(i) > 0
            forceUp(i) = motor_force1(i);
        else
            forceUp(i) = 0;
        end
    end

 % determining the amount of water pumped up
    pumped_upper_sim = volume_up(end)-volume_up(1);
    pumped_lower_sim = volume_down(end) - volume_down(1);
    avg_pumped_sim = (pumped_upper_sim - pumped_lower_sim)/2;
    pumped_int_sim = avg_pumped_sim/length(tout);
    pumped_total_sim = sum(net_flow(600:end))./(size(tout)./time);


% Calculating the potential energy
    Epot_p1 = zeros(length(tout),1);
%     Epot_p2 = zeros(length(tout),1);
%     Epot_p3 = zeros(length(tout),1);
    Ppot_sim_p1 = zeros(size(tout));
    Ppot_sim_p2 = zeros(size(tout));
    Ppot_sim_p3 = zeros(size(tout));
    h_head_sim1 = zeros(size(tout));
%     h_head_sim2 = zeros(size(tout));
%     h_head_sim3 = zeros(size(tout));

for i = 1:size(tout),
    h_head_sim_end_1 = delta_h1(end);
%     h_head_sim_end_2 = delta_h2(end);
%     h_head_sim_end_3 = delta_h3(end);
    h_head_sim1(i) = delta_h1(i);
%     h_head_sim2(i) = delta_h2(i);
%     h_head_sim3(i) = delta_h3(i);
```

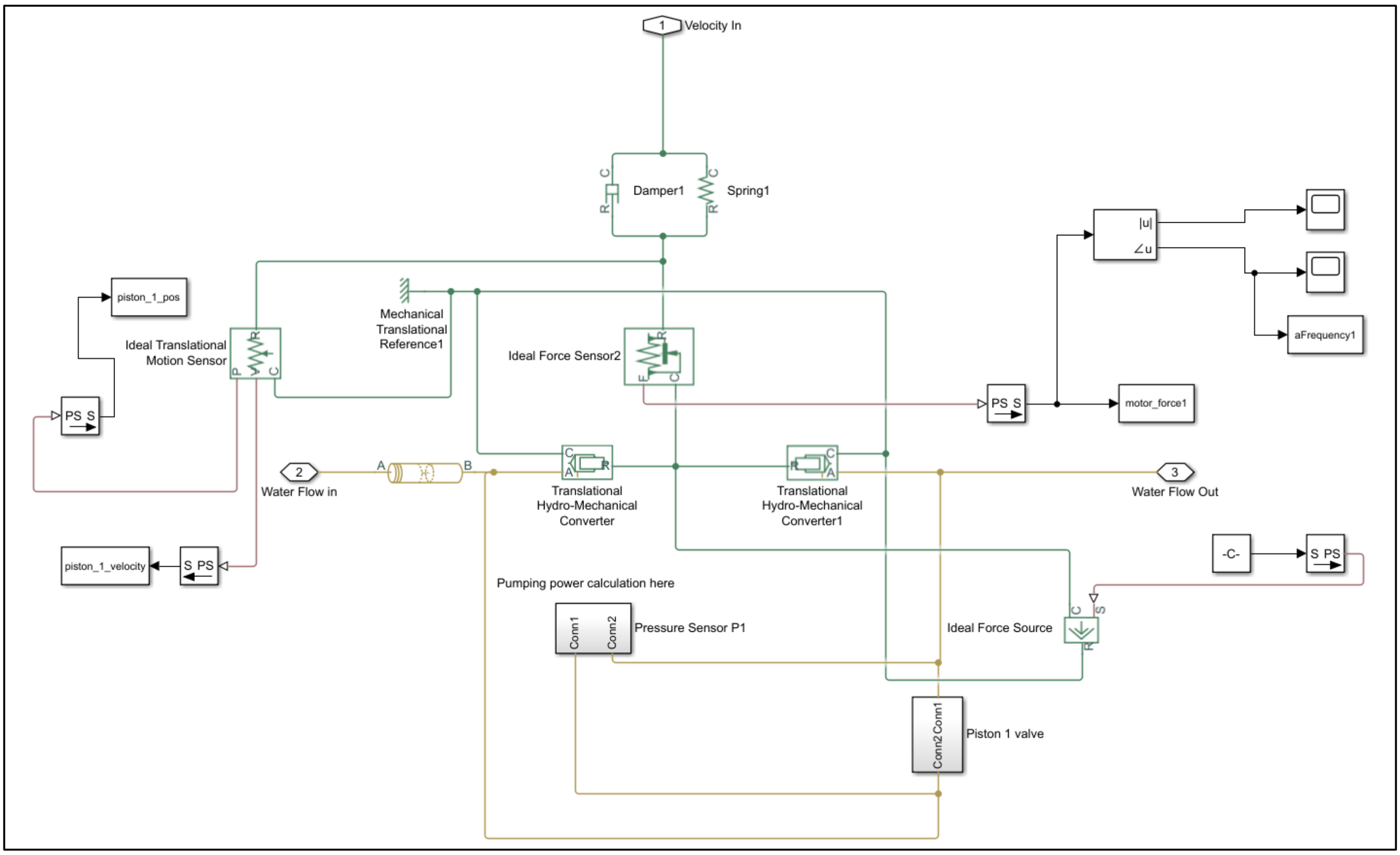
```
    Epot_p1(i) = rho*g*(sum(h_head_sim1)/length(tout))*pumped_int_sim;
%     Epot_p2(i) = rho*g*(sum(h_head_sim2)/length(tout))*pumped_int_sim;
%     Epot_p3(i) = rho*g*(sum(h_head_sim3)/length(tout))*pumped_int_sim;
    Ppot_sim_p1(i) = rho*g*h_head_sim1(i);
%     Ppot_sim_p2(i) = rho*g*h_head_sim2(i);
%     Ppot_sim_p3(i) = rho*g*h_head_sim3(i);
end

    Epot1 = Epot_p1;
%     Epot2 = Epot_p2;
%     Epot3 = Epot_p3;

     Epot = sum(Epot_p1) %+sum(Epot_p2)+sum(Epot_p3);

    avg_pumped_sim = abs(volume_up(end) - volume_up(1));
    max_pumped_sim = max(position_motor)*(pi*piston_1_rad^2)*(strokes) +
...
         max(position_motor)*(pi*piston_2_rad^2)*(strokes) + max(posi-
tion_motor)*(pi*piston_3_rad^2)*(strokes);
    vol_efficiency_sim = (avg_pumped_sim/max_pumped_sim)*100;

     Ppump_sim = (piston_1_velocity .* motor_force1) + (piston_2_velocity
.* motor_force2) + ...
        (piston_3_velocity .* motor_force3);
    Ppot_p1_end = rho*g*delta_h1(end);
    Ppot_p2_end = rho*g*delta_h2(end);
    Ppot_p3_end = rho*g*delta_h3(end);
    Epump = trapz(tout,Ppump_sim);

    tot_eff = Epot / Epump * 100;
     mec_eff = tot_eff / vol_efficiency_sim * 100;
%     mec_eff = vol_efficiency_sim / tot_eff * 100;

    % Print Results
if disp_data > 0

        disp(['-------------------------------------------------------
']);
        disp(['|          Results Simulink Model          |']);
        disp(['-------------------------']);
        disp(['Average Max displacement = ',num2str(max(position_motor)),'
m']);
    %     disp(['Average Wave Period = ',num2str(mean(Aperiod)),' Sec']);
    %     disp(['Upper level start = ',num2str(mean(h_upper_start)),' m']);
    %     disp(['Lower level start = ',num2str(mean(h_lower_start)),' m']);
        disp(['-------------------------']);
        disp(['Total Pumping energy = ',num2str(Epump),' J']);
        disp(['Total Potential energy = ',num2str(Epot),' J']);
        disp(['Avg. Pumping energy per stroke = ',num2str(Epump/strokes),'
J']);
        disp(['Avg. Potential energy per stroke = ',num2str(Epot/strokes),'
J']);
        disp(['-------------------------']);
        disp(['Volumetric Efficiency = ',num2str(vol_efficiency_sim),'
%']);
        disp(['Mechanical Efficiency = ',num2str(mec_eff),' %']);
        disp(['Total Efficiency = ',num2str(tot_eff),' %']);
else
    disp(['Nothing to show here'])
end
```

Appendix 5. Simulink model: MPPmodel_V2

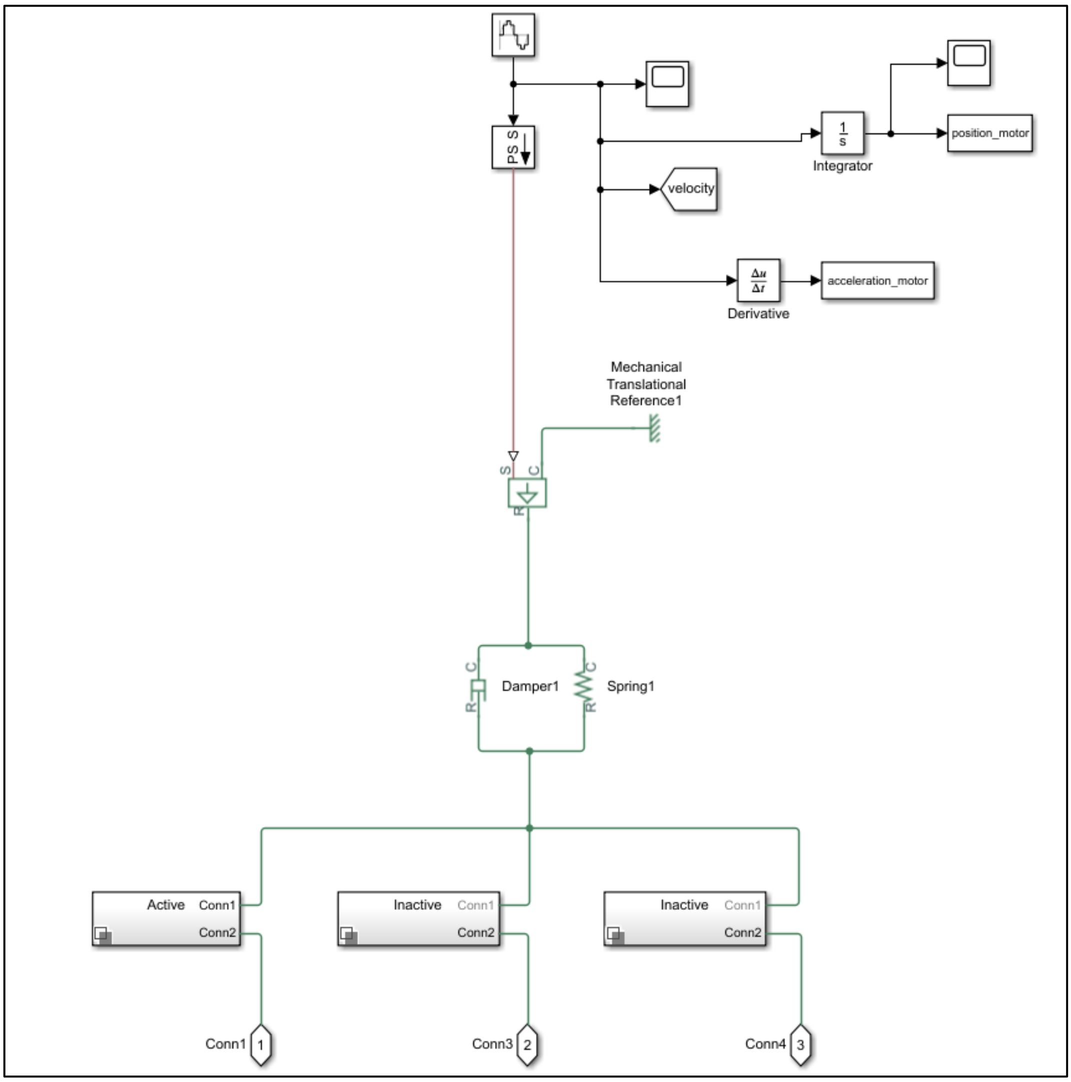



Water in
Water out P2
Water out P1
Water out P3
pipe 1
pipe 2
pipe 3
pipe 4
pipe 5
pipe 6
pipe 1.1
pipe 1.2
pipe 1.3
pipe 1.4
pipe 1.5
pipe 2.1
pipe 2.2
pipe 2.3
pipe 2.4
pipe 2.5
pipe 2.6
pipe 2.7
pipe 3.1
pipe 3.2
pipe 3.3
pipe 3.4
pipe 3.5
pipe 3.6
pipe 3.7
pipe 3.8
Butterfly Val ve
Butterfly Val ve


p_cv1
PS S
Hydraulic Pressure Sensor
CV Water Inlet
Check Valve1
Check Valve2
Check Valve3
Check Valve4
Check Valve5
Check Valve6
Check Valve7
Check Valve8
Check Valve9
Check Valve10
Check Valve11
Check Valve12
PS S
flow_cv1_total
Hydraulic Flow Rate Sensor
CV Water Outlet



Velocity In
Damper1
Spring1
piston_1_pos
Mechanical Translational Reference1
Ideal Translational Motion Sensor
Ideal Force Sensor2
|u|
∠u
aFrequency1
motor_force1
Water Flow in
Translational Hydro-Mechanical Converter
Translational Hydro-Mechanical Converter1
Water Flow Out
piston_1_velocity
-C-
Pumping power calculation here
Pressure Sensor P1
Ideal Force Source
Piston 1 valve



Integrator
position_motor
velocity
Derivative
acceleration_motor
Mechanical Translational Reference1
Damper1
Spring1
Active
Inactive
Inactive
Conn1
Conn3
Conn4

Appendix 6. Logbook

Log book

Bachelor Integratieopdracht   Semester 2 2017-2018

Name: Lennard Hut   Studentnumber: S2718960   Email: lennard.hut@gmail.com

Title of project: *Investigating the dynamic behavior of a multi-piston pump*   Supervisor: A. Vakis

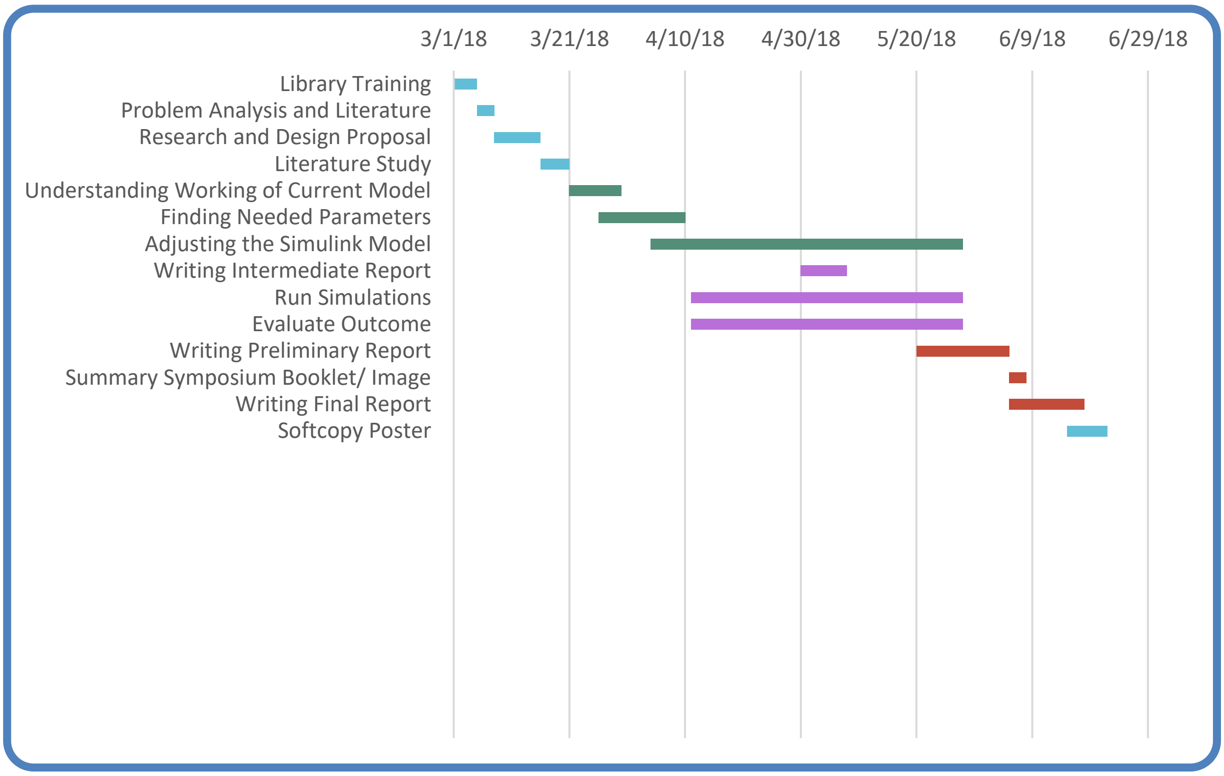
Note: the Nestor site of the Ba IP Sem 2 provides a calender of compulsory meetings and recommended activities.

number	Date	Time		Activity
1	21/02/2018	13:00	14:00	Kick off
2	21/02/2018	14:00	15:00	Project preference
3	23/02/2018	13:45	14:00	Make appointment with supervisor
4	25/02/2018	21:00	22:00	Finding information about Multi-piston Pump
5	25/02/2018	21:00	22:00	Making a list with questions for first meeting
6	26/02/2018	21:15	21:45	Literacy Test on Nestor
7	28/02/2018	14:30	15:00	First supervisor meeting
8	28/02/2018	21:15	22:30	Making Planning for First weekly meeting
9	28/02/2018	22:30	22:45	Describing the project and outline
10	01/03/2018	11:00	13:00	Workshop Library Training
11	01/03/2018	13:00	14:30	Weekly Meeting 1
12	01/03/2018	14:30	15:00	Showing of the test facility.
13	01/03/2018	20:00	21:30	Working on Information Literacy assignment
14	04/03/2018	18:45	20:00	Working on Information Literacy assignment
15	05/03/2018	10:15	15:15	Working on Information Literacy assignment
16	06/03/2018	11:00	14:30	Working on PAL
17	07/03/2018	13:00	14:30	Meeting for explanation of models
18	07/03/2018	19:30	23:15	Working on PAL
19	08/03/2018	10:00	11:00	Working on Pal
20	08/03/2018	13:00	14:30	Weekly Meeting 2
21	09/03/2018	11:00	13:00	Workshop RDP
22	12/03/2018	10:30	14:00	Working on RDP/ Revising Pal
23	13/03/2018	11:00	12:15	Meeting with Y.Wei and R.Zaharia
23	13/03/2018	19:00	22:30	Working on RDP
24	14/03/2018	10:45	17:30	Working on RDP
25	15/03/2018	13:00	13:30	Weekly Meeting 3
26	15/03/2018	14:00	16:30	Working on RDP
27	16/03/2018	08:30	11:00	Working on RDP
28	28/03/2018	12:00	19:00	Working on literature study
29	29/03/2018	13:00	14:30	Weekly meeting 4
30	31/03/2018	20:00	22:00	Working on literature study
31	01/04/2018	17:00	21:00	Working on literature study
32	14/04/2018	14:00	17:30	Rewriting parts of RDP
33	15/04/2018	11:30	17:00	Rewriting parts of RDP
34	17/04/2018	14:00	16:00	Lecture on Academic writing and Block 4

35	21/04/2018	15:00	17:00	Rewriting parts of RDP
36	23/04/2018	13:00	16:30	Rewriting parts of RDP
37	24/04/2018	12:30	16:30	Rewriting parts of RDP
38	25/04/2018	12:00	13:00	Rewriting parts of RDP
39	25/04/2018	14:00	16:00	Workshop Academic writing
40	25/04/2018	18:00	21:45	Rewriting current model explanation
41	26/04/2018	13:00	13:30	Weekly meeting 5
42	28/04/2018	11:00	15:30	Investigating parameters
43	28/04/2018	17:15	19:00	Investigating parameters
44	30/04/2018	11:30	17:00	Working on literature study
45	01/05/2018	12:00	17:30	Working on literature study
46	02/05/2018	11:45	16:30	Modeling in Simulink/Matlab
47	03/05/2018	13:00	14:30	Weekly Meeting 6
48	03/05/2018	14:30	15:15	Meeting with Marijn
49	04/05/2018	11:15	17:00	Calculating parameters
50	04/05/2018	20:30	22:00	Calculating parameters
51	05/05/2018	11:00	15:00	Calculating parameters
52	07/05/2018	11:15	15:00	Calculating param./modeling/ Intermediate
53	07/05/2018	15:45	17:00	Calculating param./modeling/ Intermediate
54	07/05/2018	23:15	02:30	Intermediate report writing
55	08/05/2018	11:15	11:30	Handing in IR
56	08/05/2018	12:00	15:45	Working on report and parameters
57	14/05/2018	12:30	17:00	Working on report and parameters
58	15/05/2018	11:00	15:00	Working on report and parameters
59	15/05/2018	16:00	16:30	Meeting with Marijn for check valve par.
60	16/05/2018	10:30	14:30	Modeling in Simulink/Matlab
61	16/05/2018	16:00	17:00	Modeling in Simulink/Matlab
62	17/05/2018	11:00	13:00	preparing presentation
63	17/05/2018	13:00	14:30	Weekly Meeting 7
64	18/05/2018	15:30	18:00	Modeling in Simulink/Matlab/Parameters
65	19/05/2018	12:00	20:30	Modeling in Simulink/Matlab/Parameters
66	19/05/2018	22:15	23:00	Modeling in Simulink/Matlab/Parameters
67	20/05/2018	12:30	21:30	Modeling in Simulink/Matlab/Parameters
68	21/05/2018	12:00	16:30	Calculations CV and PV
69	21/05/2018	20:00	21:15	Calculations CV and PV
70	22/05/2018	11:00	11:30	Meeting with supervisor for CV&PV
71	22/05/2018	13:30	14:30	Calculations CV and PV
71	22/05/2018	18:00	22:00	Calculations CV and PV
72	23/05/2018	11:00	11:30	Meeting Marijn
73	23/05/2018	12:15	15:30	Writing, Calculating and Modeling
74	23/05/2018	18:00	22:00	Writing, Calculating and Modeling
75	24/05/2018	13:00	15:15	Weekly Meeting 8
76	25/05/2018	11:00	12:30	Intermediate Report Presentations
77	25/05/2018	18:00	00:00	Writing, Modeling
78	26/05/2018	21:00	23:00	Writing, Calculating and Modeling

79	27/05/2018	11:00	15:00	Writing, Calculating and Modeling
80	27/05/2018	20:30	21:45	Writing, Calculating and Modeling
81	28/05/2018	10:30	15:15	Writing, Calculating and Modeling
82	28/05/2018	16:00	19:45	Writing, Calculating and Modeling
83	28/05/2018	21:00	22:30	Writing, Calculating and Modeling
84	29/05/2018	11:00	16:00	Writing, Calculating and Modeling
85	29/05/2018	20:00	23:00	Writing, Calculating and Modeling
86	30/05/2018	10:00	12:30	Meeting Robert
87	30/05/2018	13:45	15:45	Writing, Calculating and Modeling
88	30/05/2018	18:30	20:15	Writing, Calculating and Modeling
89	30/05/2018	21:00	01:30	Writing, Calculating and Modeling
90	05/06/2018	01:00	01:15	Handing in Preliminary report
91	05/06/2018	11:00	14:30	Actuation mechanism
92	05/06/2018	15:00	15:15	Meetig marijn about output actuator
93	05/06/2018	20:00	21:30	writing final report
94	06/06/2018	10:30	16:00	writing final report/symposium summary
95	07/06/2018	13:00	15:00	Weekly Meeting 9
96	07/06/2018	18:00	22:00	Symposium Image / Symposium summary
97	08/06/2018	11:00	11:15	Handing in symposium image/summary
98	08/06/2018	13:30	16:00	Meeting with Yanji
99	08/06/2018	20:30	23:00	writing final report
100	09/06/2018	10:00	21:00	writing final report
101	11/06/2018	12:45	18:45	writing final report
102	11/06/2018	21:00	22:00	writing final report
103	12/06/2018	15:00	16:00	Presentation lecture
104	12/06/2018	17:45	19:45	writing final report / working on poster
105	12/06/2018	21:30	22:30	Poster
106	13/06/2018	13:00	15:00	Poster workshop
107	14/06/2018	13:00	14:15	Weekly Meeting 10
108	14/06/2018	14:45	19:45	Enlarged graphs/poster/finalizing coding
109	14/06/2018	20:45	22:00	finalizing coding/ damping coefficient
110	15/06/2018	11:45	12:45	Discussion with Yanji
111	15/06/2018	18:00	20:00	Finalizing / implementing feedback Yanji
112	16/06/2018	11:00	17:00	Finalizing report
112	16/06/2018	18:00	19:30	Finalizing report
113	17/06/2018	11:30	18:15	Finalizing report
114	17/06/2018	20:45	22:00	Finalizing report
115	17/06/2018	23:00	01:00	Finalizing report
116	17/06/2018	10:00	11:30	last check

Appendix 7. Gantt Chart of activities (Planning block II)

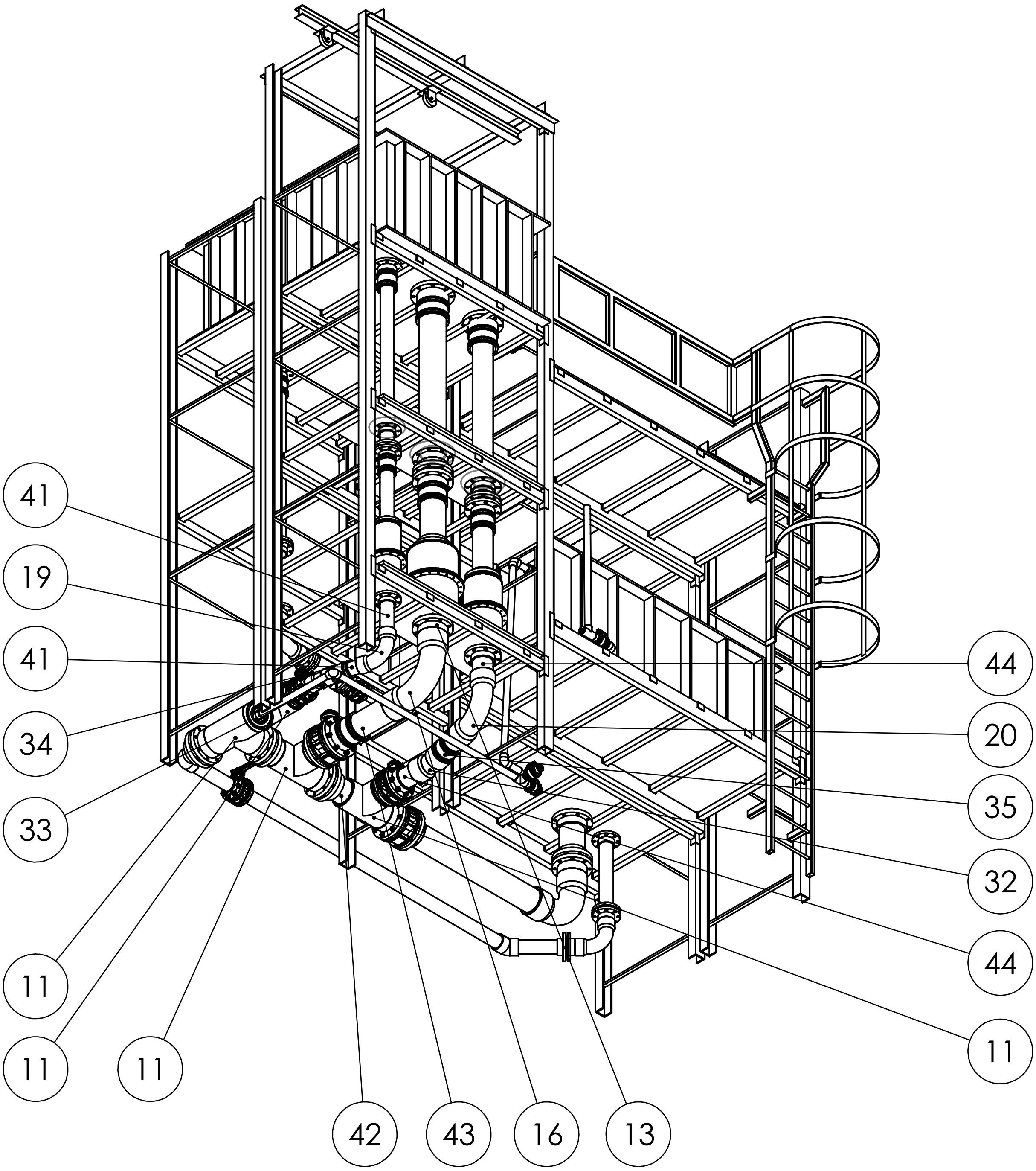


Appendix 8. Bill of Material (BOM) and technical drawing

ITEM NO.	PART NUMBER	QTY.
1	Diepteligger110	46
2	Frama480	4
3	Frama600	2
4	Ligger1850	12
5	Ligger2700Ligt	6
6	Ligger2700Zwaar	4
7	watertank200x250x100	1
8	watertank180x220x100	1
9	dn200x150cm	1
11	Assem_TEE	1
11	2_01_200	3
12	3_72_200	9
13	3_72_201	9
14	3_73_200	4
15	6_78_200	2
16	1_05_200	2
17	3_05_197	3
18	3_05_200	1
19	1_05_110	3
20	1_05_160	1
21	6_78_110	3
22	6_78_160	1
23	2_25_110	1
24	dn110x50cm	2
25	3_72_110	13
26	3_72_111	13
27	3_73_110	6
28	2_30_110	1
29	dn110x287cm	1
30	dn110x43cm	1
31	dn110x25cm	1
32	1_14_160	1
33	1_14_110	1
34	FlexKoppeling110	3
35	FlexKoppeling160	2
36	FlexKoppeling200	3

ITEM NO.	PART NUMBER	QTY.
37	3_73_225	1
38	3_73_125	1
39	FlexKoppeling125	1
40	FlexKoppeling225	1
41	dn110x35cm	2
42	dn200x26cm	1
43	dn200x66cm	2
44	dn160x26cm	2
45	dn160x40cm	1
46	2_25_050	8
47	3_05_110	1
48	dn50x80cm	1
49	dn50x200cm	1
50	2_01_050	3
51	dn110x168cm	1
52	6_15_050	5
53	3_10_048	3
54	5_17_050	4
55	DN50x15cm	5
56	DN50x120cm-acrylic	2
57	DN50x180cm	2
58	wooden_plate	3
59	Ladder with safety cage	1
60	safety-heck	1
61	flange_110mm	1
62	pipe_100mmh9	1
63	flange_160mm	1
64	pipe_140mmh9	1
65	flange_200mm	1
66	pipe_190mmh9	1
67	3_72_160	5
68	3_72_161	5
69	dn110x18cm	1
70	dn160x15cm	1
71	CVdn100-200mm	2
72	CVdn140-300mm	2

ITEM NO.	PART NUMBER	QTY.
73	CVdn200-400mm	2
74	dn110x54cm	1
75	dn160x54cm	1
76	dn200x54cm	1
77	guideplatecylinders_upper	1
78	guideplatecylinders_lower	1
79	2_03_110	1
80	steeltop_part1	2
81	steeltop_part2	3
82	steeltop_part3	1
83	steeltop_pulley	2



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL:

WEIGHT:

TITLE: Numbers corresponding to BOM for the inflow pipes

DWG NO. Design_v4

A4

SCALE:1:100

SHEET 1 OF 1