AALBORG UNIVERSITY

STUDENT REPORT

P4

Flow Control

- and general understanding about pumps -

Students:
Daniel Frederik Busemann
Razvan-Vlad Bucur
Troels Ulstrup Klein

Supervisors:
Petar Durdevic Løhndorf
Simon Pedersen

May 28, 2018

LaTeX was used for typesetting this report, MatLab and Simulink were used for designing and verifying the controller and GitHub [1] for collaborating as a group.

Electronics and Computer Engineering
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:

Theme:
Control Theory

Project Period:
Fall Semester 2017

Project Group:
ED4-1-F18

Participant(s):
Daniel Frederik Busemann
Razvan-Vlad Bucur
Troels Ulstrup Klein

Supervisor(s):
Petar Durdevic Løhndorf
Simon Pedersen

Copies: 1

Page Numbers: 59

Date of Completion:
May 28, 2018

Abstract:

This report describes two versions of mathematical modelling for pumping systems. All data used for the coefficients was gathered on a setup of three identical pumps in parallel, but only one pump was used at a time.
In the early stages of the project the goal was to use all three pumps for energy efficiency, this was however not done because of the limited time frame.
Instead a PID controller was implemented and tuned with the Ziegler Nichols method. The coefficients were based on experimental data gathered on aforementioned setup.
Additionally a static model of the head relative to the flow was developed.
The final controller implementation is done with Simulink® Real-Time™, on a target PC.

Preface

This report was made by three students from Aalborg University Esbjerg attending the 4th semester of the Electronics and Computer Engineering course. From this point on, every mention of **we, the group** or **the authors** refers to the three co-authors listed below.

All resources produced for this project can be found on the GitHub repository [1]. Selected resources are also in the appendix.

Aalborg University, May 28, 2018

Daniel Frederik Busemann
<dbusem16@student.aau.dk>

Razvan-Vlad Bucur
<rbucur16@student.aau.dk>

Troels Ulstrup Klein
<tklein11@student.aau.dk>

Contents

Acronyms & Nomenclature

Symbol	**Definition**	**Unit**
H	Head	*bar*
MFM	Magnetic Flow Meter	
MIMO	Multiple Input Multiple Output	
ML	MATLAB®	
NPSH	Net Positive Suction Head	*bar*
OL	Open-loop	
OS	Operating System	
P	Power	W
PC	Personal Computer	
QDM	Quarter Decay Method	
SISO	Single Input Single Output	
SL	Simulink®	
SLRT	Simulink® Real-Time™	
USM	Ultimate Sensitivity Method	
ΔP	Pressure difference	*bar*
η	Efficiency	%
ρ	Density	$\frac{kg}{m^3}$
ω	Rotational speed	$\frac{rad}{s}$

	Subscripts
cl	Controller
el	Electric
hyd	Hydraulic
ref	Reference
tot	Total

	Prescripts
Δ	Change

Chapter 1

Introduction

The human body consists of 60% water [2]. To remain healthy, it is recommended to consume between 2,7 l and 3,7 l of water every day [3]. This intake of water comes from both drinks and food. The water is often added during growth, processing, or preparation of food. For drinks to reach the consumer it has to be transported, in the form of bottles, tanks, or through pipelines.

The ability to precisely control liquids is also very important in many industries. Examples range from the oil and gas sector [4] over breweries [5], dairy plants [6] to waste water treatment plants [7]. In every commercial use, it is important to keep cost low. The lifetime cost of any system depends both on the initial cost, but especially the running cost [8]. The running cost of controlling flow depends largely on the efficiency of the pumps used [9].

Prior work in the field shows, that efficiency can be improved by changing from a single pump to multiple pumps, when the scheduling of those are optimally controled [10]. The mentioned research only focuses on controlling pressure, and it would therefore be interesting to investigate, whether this also is true for controlling flow.

The focus of this report therefore lays on controlling a constant flow, after intensive modelling of the system at hand.

Reading Guide

In Chapter 3 the physical setup gets explained, giving a short introduction to all components relevant for this report. Chapter 5 and 6 focus on the mathematical modelling of those components and their interdependencies. Focus is mainly put on the relations between Flow, Head and Efficiency according to a given Pump Speed. Different modelling techniques are explained and evaluated, alongside a very brief introduction to MATLAB®, Simulink® Real-Time™ and xPC Target, the software used for this process. Chapter 4 explains how the experiments were conducted and what results were achieved. Chapter 7 covers how the model was used to design a controller fulfilling our requirements stated in Chapter 2. Chapters 8 and 9 summarize the findings from this work, propose possible future work and pose a conclusion to the work done.

Chapter 2

Problem Description

2.1 Problem Description

This project is about different mathematical modelling approaches for a system consisting mainly of three pumps. The modelling will be used to develop a controller governing the total flow Q_{tot}. The system was already fully functionally available in our university. No alterations on the setup were possible to complete our project, since the system was simultaneously used by two other groups. As control inputs were available the individual speed of each pump $\omega P_{1,2,3}$ and the valve opening of one control valve CV_1.

Sensed outputs on this system are individual flow, individual differential pressure, pressure over CV_1 and individual power consumption ($P_e l$). Individual meaning separate for each pump.

A Piping and Instrumentations Diagram (P&ID) of the system is available in Appendix A.

2.2 Control Methods

Based on the properties of a system two different approaches to controlling it can be applied. Classical Control refers to the use of transfer functions and generally full output feedback. This is generally preferred for simpler systems, mostly Single Input Single Output (SISO) systems, because one transfer function is needed for all connections between each input and output. A very common and simple control scheme for these systems is the PID control, explained later in Chapter 7. [11]

Multiple Input Multiple Output (MIMO) systems are often modelled and controlled using state-space modelling, where all inputs are combined into one input vector and the plant is described as a combination of three to four matrices. In modern control this approach is commonly used in combination with full state feedback, where instead of the output an intermediate product (the states) are used for full-state feedback. Since we are not using this approach in our project no detailed explanation will be given in this report.

2.3 Problem Delimitation

Our system can be seen as a SISO, MIMO or something in between, depending on what is to be controlled. If only a single pump is considered and only the output flow to be controlled, it is a SISO system. Using multiple pumps and controlling for example Q and H would effectively make it a MIMO system.

In this project, we decided to use the system as a SISO system, controlling the total flow of all three pumps by regulating a single pump. Primary goals are therefore:

- Creation of a dynamic model for one pump
- Design of a PID controller for Q
- Tuning of said PID controller

In addition we also had some secondary goals, which we deemed not necessary for successful completion of the project, but nice extras.

- Creation of a static model for one pump
- Creation of a static model for multiple pumps
- Design of a controller for the flow, taking efficiency into account

The dynamic model will be useful to tune the PID controller, and give us a deeper insight into the working of the pumps. It would also make it possible to simulate and theoretically test different controller designs.

2.3.1 Requirements

To have a goal while tuning the PID controller, we gave ourselves the following requirements.

- Maximum Overshoot $M_p = 0\%$
- Steady-state error $e_{ss} \leq 1\%$

We chose not to put a requirement on the settling time t_s, because we could not initially estimate how the system would behave, since we had no previous experience with pumps.

Chapter 3

Physical Setup

This chapter explains how the setup is built. Going into detail about the pump type used, centrifugal pumps, since they determine the major dynamics of the system. The other components, pipes, valves and sensors are also explained, but in less detail.

To give a general overview of the setup, Figure 3.1 shows the front view of the setup, with all three pumps (red and black) and their individual sensors (blue) visible. Also visible are the control valve, the target computer, parts of the electrical wiring and parts of the piping.

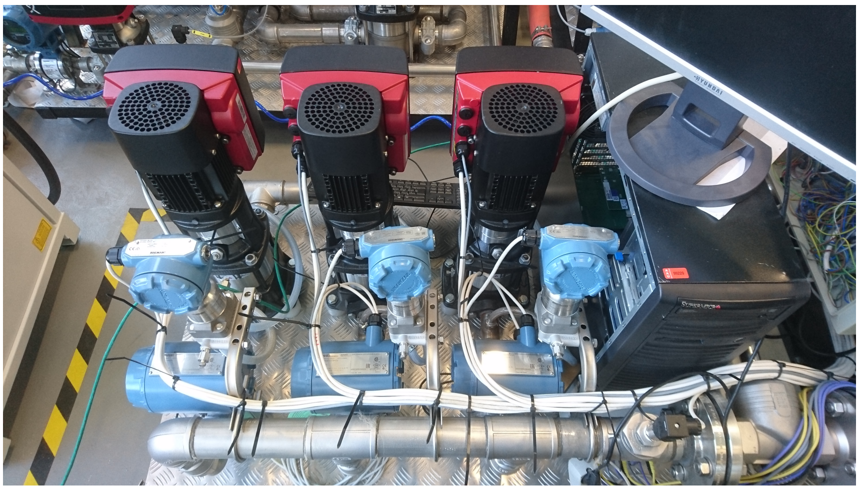

Figure 3.1: Photo of the setup

Not visible on this photo are the power sensor and the water tank.

A schematic overview of the system is available in Appendix A.

This setup was already available for use with student projects and two other groups used it simultaneously, which is why no alterations were taken.

3.1 Centrifugal Pumps

A pump is a device used to move liquid through a piping system and to raise the pressure of the liquid. The rise in pressure is often necessary for processes upstream or to overcome a rise in the pipeline. We will focus on explaining and describing only centrifugal pumps, since this is the type of pump present in our setup.

3.1.1 Principle

In 1689, physicist Denis Papin invented the centrifugal pump. It is the most commonly used type of pump, due to its simple construction, relative low cost, reliability and quiet operation.

The pump is built on a simple principle: Liquid is led to the impeller hub and by means of the centrifugal force it is flung towards the periphery of the impeller [12].

Figure 3.2 shows two cross sections of a centrifugal pump, showing the flow of liquid through the pump.

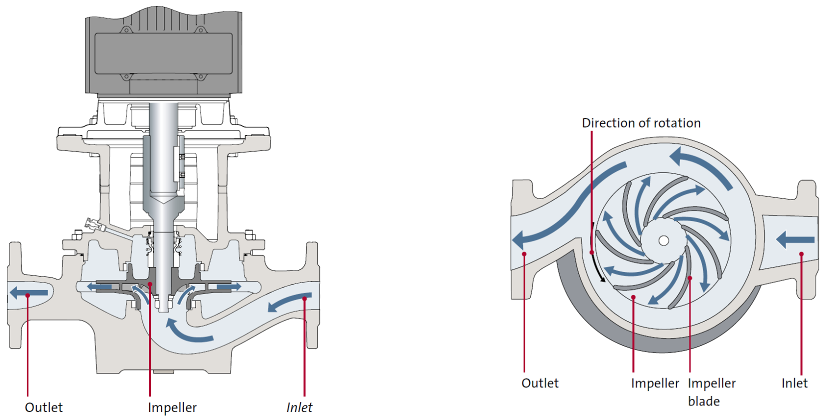

Figure 3.2: Centrifugal Pump [12]

The fluid is sucked into the impeller at the impeller eye and flows through the impeller channels formed by the blades between the shroud and hub. The blades of the rotating impeller transfer energy to the fluid by increasing velocity and pressure. [13]

The design of the impeller depends on the requirements for application, pressure and flow. The impeller is the primary component determining the pump performance. Pumps variants are often created only by modifying the impeller.

Changing the impeller size of a pump does not influence all output characteristics equally, but the change can be modelled with the Affinity Laws.

3.1.2 Affinity Laws

Affinity laws are mathematical relationships that provide a way to estimate the changes in performance of a pump, as a result of a change in one of the basic pump variables. In it's simplest form, the term law, means a principle that has been proven true for all cases [14].

Equations 3.1 show the relations between a change in motor speeds N to flow Q, head H and power P.

$$\frac{Q_1}{Q_2} = \left(\frac{N_1}{N_2}\right) \qquad \frac{H_1}{H_2} = \left(\frac{N_1}{N_2}\right)^2 \qquad \frac{P_1}{P_2} = \left(\frac{N_1}{N_2}\right)^3 \tag{3.1}$$

[9]

Similarly, the equations 3.2 show the relation between a change in impeller diameter D to Q, H and P.

$$\frac{Q_1}{Q_2} = \left(\frac{D_1}{D_2}\right) \qquad \frac{H_1}{H_2} = \left(\frac{D_1}{D_2}\right)^2 \qquad \frac{P_1}{P_2} = \left(\frac{D_1}{D_2}\right)^3 \tag{3.2}$$

[9]

3.1.3 Performance Curves

Performance curves are a very widespread way to compare pumps and estimate what pump is needed in a specific application. We included them in this report, because they also help understanding the relations of Q, H and $\omega P/N$.

Pump Head Curve

A QH-curve or pump curve defines the head as a function of the flow. The flow is the rate of the fluid going through the pump. It is generally stated in cubic meters per hour ($\frac{m^3}{h}$). Figure 3.3 shows a typical pump head curve. Interesting to notice is, that the pressure decreases quadratically with the flow. This can be explained with the Affinity laws mentioned earlier.

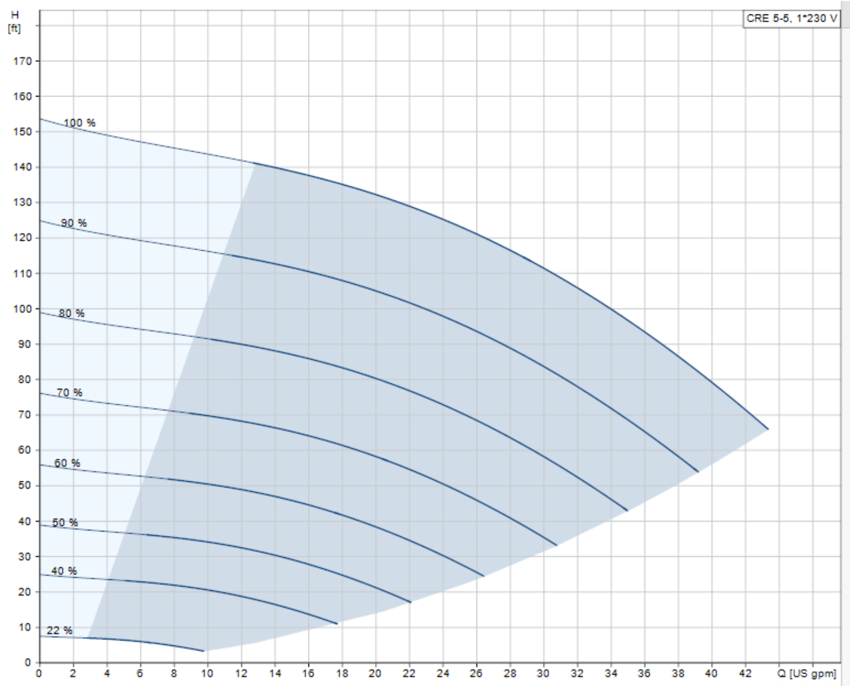

Figure 3.3: Typical Pump Head Curve [15]

Power Consumption Curve

The power consumption P_{el} is another key factor, when choosing a pump for an application. The typical between Q and P_{el} can be seen in Figure 3.4. Generally, the power consumption increases with the flow.

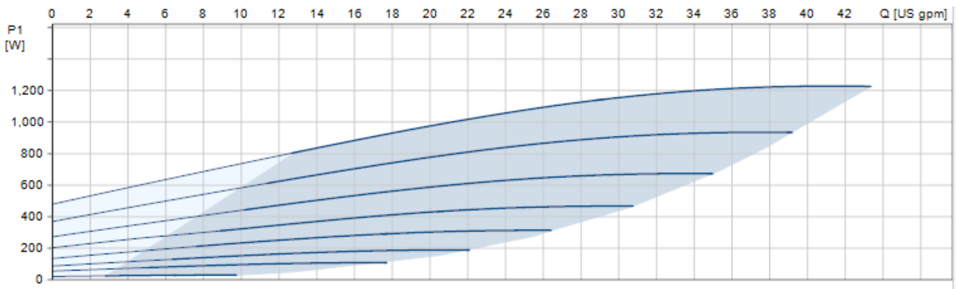

Figure 3.4: Typical Power Consumption Curve [15]

Efficiency Curve

The efficiency η of a pump is the relation between the power delivered from the pump to the water P_{hyd} and P_{el}. Figure 3.5 shows a typical efficiency curve of a pump. The yellow dot represents the chosen operating point of the pump. The red dot, shows the efficiency of the pump, at that specific operating point.

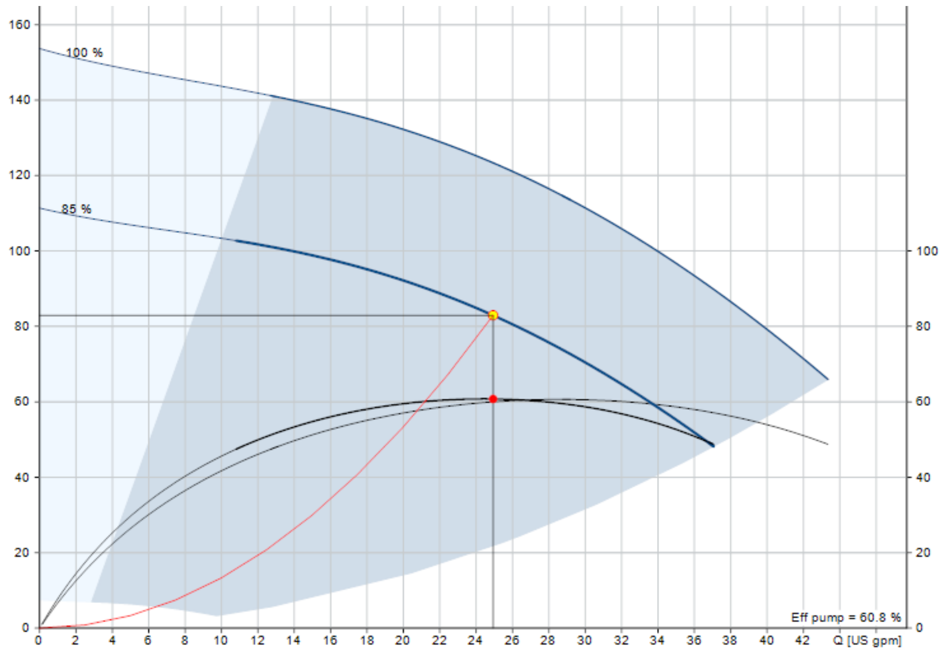

Figure 3.5: Typical Efficiency Curve (y-axis: H [ft]) [15]

Noticeable here is, that η does not linearly correlate to P_{el}.

NPSH Curve

The Net Positive Suction Head (NPSH) is the minimum required pressure that has to be present at the inlet to avoid cavitation. The NPSH increases with the flow. Figure 3.6 shows a typical NPSH Curve. It can be observed that for a flow of approximately 32 gpm a NPSH of approximately 8 feet is required.

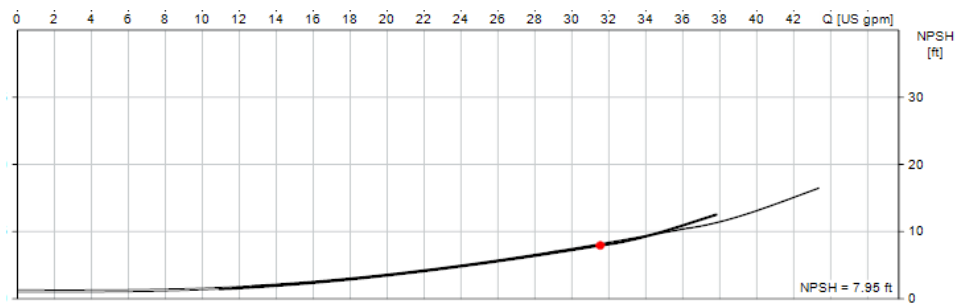

Figure 3.6: Typical NPSH Curve [15]

3.2 Pipes

Pipes are a way of transporting liquids or gasses, inside a controllable environment. They are used to interconnect the pumps and the tank and other peripherals. One common analogy compares them to wires in electrical circuits.

Based on their diameters, material and shape, they introduce resistance to the flow of the pumped medium. Staying with the analogy to electrical circuits, this can be compared to the cross sectional area and the specific resistance of a wire material.

3.3 Valves

A valve is a device used to regulate the flow of a gas or liquid through a piping system. Valves can be actuated by different means, such as air pressure, electric motors or rotary handles. The valve present in our setup is a Bürkert control valve. In our project, we are not using the valve for regulatory purposes, but only to simulate disturbances and resistance in the system.

3.4 Target PC

A target PC is often used when a setup has to be run in real-time. In our setup it is a desktop PC running xPC target, a real-time capable OS. It is equipped with a PCI card to collect the data from all sensors. [16] In other cases a different PC setup might be used, with application specific hardware. Because the target PC is not a key part of this project, but merely a tool, we will not go into detail about it.

3.5 Sensors

When we started the project, the setup was already equipped with several sensors. This section will briefly explain what sensors are used.

Flow Meter

Downstream from each pump, a magnetic flow meter (MFM) [17] is installed to measure the flow. Their digital gauge was used to calibrate the sensors data output to cubic meters per hour ($\frac{m^3}{h}$).

Power Sensor

A power sensor [18] was used to closely monitor the power delivered to the pumps. This power works like a digital multimeter, measuring voltage U and current I.

From those measurements it calculates the power P. This sensor has a digital display, where U $[V]$, I $[A]$ and P $[kW]$ are displayed, which we used to calibrate the output to kW.

ΔPressure Sensor

Pressure sensors [19] are installed across each pump, measuring the pressure difference between the inlet and the outlet of the pump. Because they didn't have a visual representation of their measurements and all our previous calculations of sensor calibration fit with previous work done on the setup [20], we chose to rely on previous calibration and inherited their conversion. After this conversion they output data is measured in bar.

Chapter 4

Experiments and Lab Work

To relate the theory behind controlling pumps to the real world, we conducted different experiments. This helped to gain knowledge about the pump system, to test controller implementations, and to gather the data needed for tuning of the controller.
All figures shown in this chapter are also available in Appendix D.

4.1 Data Acquisition

The pump system is built as described in Chapter 3, with the controlling PC running xPC Target, a real-time OS for use with Simulink® Real-Time™ (SLRT).

All data was collected with the help of custom made MATLAB® (ML) scripts and SLRT models. The execution of these was done on the xPC Target OS. Specific parts of these files will be explained in this chapter, while the complete files can be found on the GitHub repository [1].

To use SLRT on xPC targets, a SL model of the system was made on the host PC, compiled, transferred over Ethernet and executed on the xPC target. After the real-time execution is finished on the target, the .dat files containing the recorded data have to be transferred back to the host, where further analysis can be done.

To automate as much of this process as possible, we created a ML script that updates and compiles the SL model, transfers it to the target, starts the execution and copies the generated .dat files to the host.

4.2 System Test

To obtain information about how the system reacts under different conditions, a test was carried out with some example conditions. From those conditions, the goal is to extrapolate an equation for the system, that can estimate the systems reaction at different conditions. To obtain the data needed, a single pump is run at different speeds, while flow resistance is varied by a choke valve, resulting in corresponding values of flow, pressure, and energy consumption measured by the sensors. The measured data was stored for further analysis such as creating pump curves.

To get an overview over the whole operating range of the setup, a very broad test was conducted. We wanted to find the correlations between pump speed, backpressure, flow and power consumption. To get reliable results, we chose to only change one variable at a time. Since the value to be changed by our controller was expected to be the pump speed $\omega P_{1,2,3}$, we decided to fix the backpressure by fixing the control valve position, and stepwise change $\omega P_{1,2,3}$. Since the three pumps in the setup are expected to be identical, the test was only run with one of the pumps.

4.2.1 Gathered Data

Several tests were done on the setup, in order to get a general overview of the system. Pump 2 was chosen at random. We have run several tests, where we have gradually changed the pump speed in order to monitor the flow, pressure and power consumption. Starting from 0% pump speed (ωP), we have increased ωP by 10% every 15 seconds. This was done in order to give the system some time to stabilize. The process was repeated for a range of valve openings. Similar to ωP, the valve was in the beginning 10% open, gradually increasing by 10% and finally reaching 100%. Several identical tests were conducted, in order to see if the values would hold.

Figure 4.1 shows the raw flow measurements, while Figure 4.2 and Figure 4.3 show the raw power and pressure measurements. Interesting to notice here is that the lowest values for Q appeared at the lowest values for CV_1, while P behaves in the opposite manner, having its highest values at the lowest CV_1.

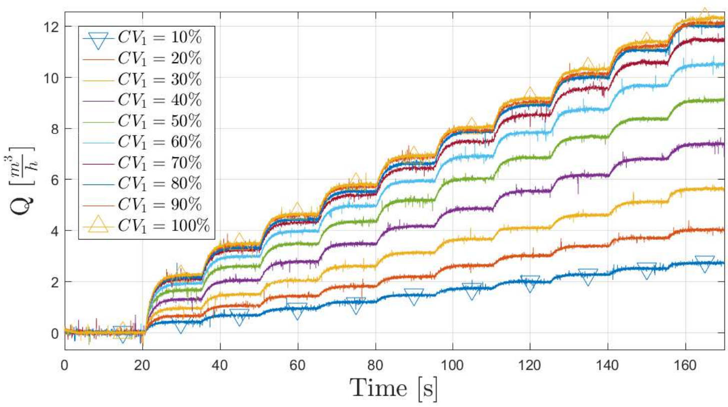

Figure 4.1: Measured Flow

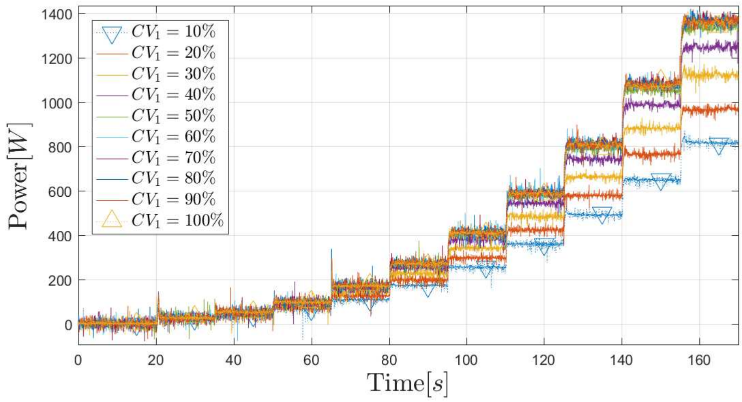

Figure 4.2: Measured Power

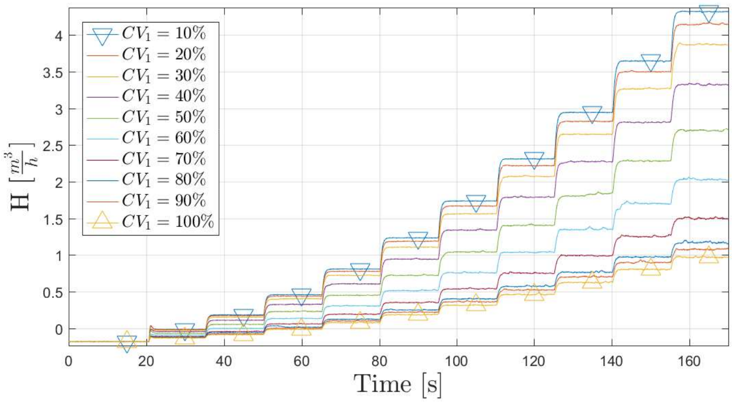

Figure 4.3: Measured Pressure

4.2.2 Data Example

To give some more insight in the testing, we wanted to show the data collected in a single run. The Figures 4.1, 4.2 and 4.3 show the collected data through several runs, each with a different CV_1. The data for all lines $CV_1 = 70\%$ in all three figures comes from the same run. While executing a run, the data for all used sensors was shown on the target in real time. To get a better understanding of how the tests were executed, we included Figure 4.4, a rebuilt live view.

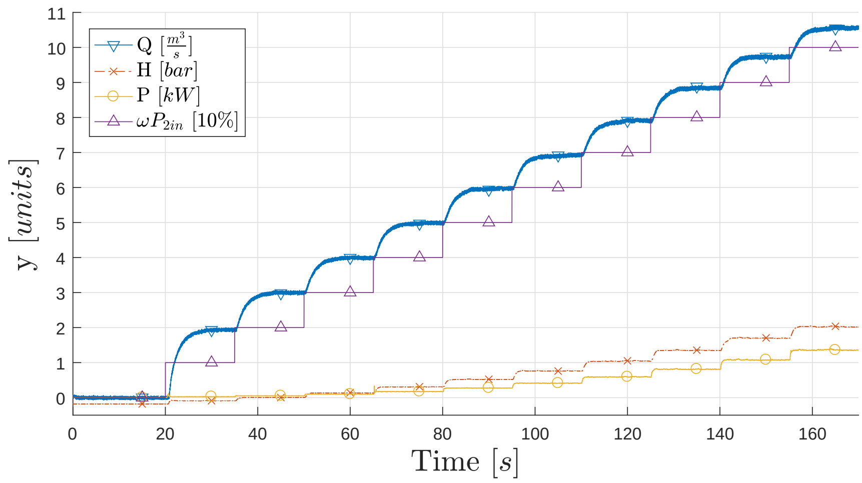

Figure 4.4: Q, H, P and stair input ωP_{2in} at $CV_1 = 70\%$

Chapter 5

Mathematical Modelling

Most physical systems can be modelled statically or dynamically. Depending on the application of the model, and therefore on the end goal of the controller, either option has some benefits and flaws.

Since our focus shifted during the project, we started out developing a static model, describing the steady state response of the system. Later on we decided to move towards a dynamic model, for easier PID-controller development.

Both modelling processes are described in this chapter.

5.1 Static Modelling

The static model explains the behaviour of the system at steady-state, i.e. when the output is settled after a step input. Static models are therefore also referred to as steady-state models. Both expressions are used interchangeably throughout this report.

Typical static models for pump systems are pump curves and system curves.

We used the data gathered through experiments to develop a model that would fit our data with some error. A model that could fit all the data without any error would have to be of a very high order and would probably model the noise better than the actual system..

We decided to use grey-box modelling with polynomial fitting. We consider our approach as grey-box rather than black-box, because we chose the degrees of the polynomials based on known and tested physical relations.

A very similar modelling process was already successfully used by Pedersen and Yang [10] on the same setup. The choice of degree for the polynomials was determined by the affinity laws [9] and supported by very good fit to the data.

ML Curve Fitting Toolbox (*cftool*) [21] was used in order to find the polynomial coefficients most accurately describing the system.

Single Pump Model

Equations 5.1 and 5.2 show a model that determines H and P, at a given Q [22]. Although the formula does not directly relate to the pump speed, it indirectly relates to it, due to the fact that only one possible pump speed ω exists for a given flow, provided the impeller diameter D remains constant (compare 3.1.2).

$$H = \bar{a}_2 \cdot Q^2 + \bar{a}_1 \cdot Q + \bar{a}_0 \tag{5.1}$$

$$P = \bar{b}_3 \cdot Q^3 + \bar{b}_2 \cdot Q^2 + \bar{b}_1 \cdot Q + \bar{b}_0 \tag{5.2}$$

The coefficients are determined by the pump characteristics and can be experimentally identified. Taking variable speed into account, the equations 5.1 and 5.2 will depend on the motor speed [22]. The pump Affinity Laws state:

$$\left(\frac{\omega_1}{\omega_2}\right)^1 = \frac{Q_1}{Q_2} \qquad \left(\frac{\omega_1}{\omega_2}\right)^2 = \frac{H_1}{H_2} \qquad \left(\frac{\omega_1}{\omega_2}\right)^3 = \frac{P_1}{P_2}$$

[9].

Assuming the pump model parameters ($\bar{a}_0$, $\bar{a}_1$, $\bar{a}_2$ and $\bar{b}_0$, $\bar{b}_1$, $\bar{b}_2$, $\bar{b}_3$) described in equation 5.1 and 5.2 are obtained at a certain speed $\bar{\omega}_0$, this results in the following equation describing the pump at any speed ω [22].

$$H(\omega) = a_0 \cdot \omega^2 + a_1 \cdot \omega \cdot Q(\omega) + a_2 \cdot Q(\omega)^2 \tag{5.3}$$

$$P(\omega) = b_0 \cdot \omega^3 + b_1 \cdot \omega^2 \cdot Q(\omega) + b_2 \cdot \omega \cdot Q(\omega)^2 + b_3 \cdot Q(\omega)^3 \tag{5.4}$$

The coefficients can be determined as follows:

$$a_0 = \frac{\bar{a_0}}{\bar{\omega}_0^2} \qquad a_1 = \frac{\bar{a_1}}{\bar{\omega_0}} \qquad a_2 = \bar{a_2}$$

$$b_0 = \frac{\bar{b_0}}{\bar{\omega}_0^3} \qquad b_1 = \frac{\bar{b_1}}{\bar{\omega}_0^2} \qquad b_2 = \frac{\bar{b_2}}{\omega_0} \qquad b_3 = \bar{b_3}$$

[22]

As observed in equations 5.3 and 5.4 the coefficients are initially multiplied by the input speed. Afterwards they are divided by the speed they were determined for, in order to take into account the variable speed [22].

Figure 5.1 shows data points connected together that represent a pump head curve. As observed in the figure, when the opening percentage of the control valve is at 10% the pressure seems to reach negative values. We believe, this arises from the fact that we were not able to properly calibrate the pressure sensors. As an alternative, we have used the gain and offsets provided to us by previous researchers that have experimented on the same setup [20]. The gains and offsets that we were able to determine for Flow and Power Consumption, respectively, were also consistent with the values determined by them. This is the reason we decided to use the values provided to us.

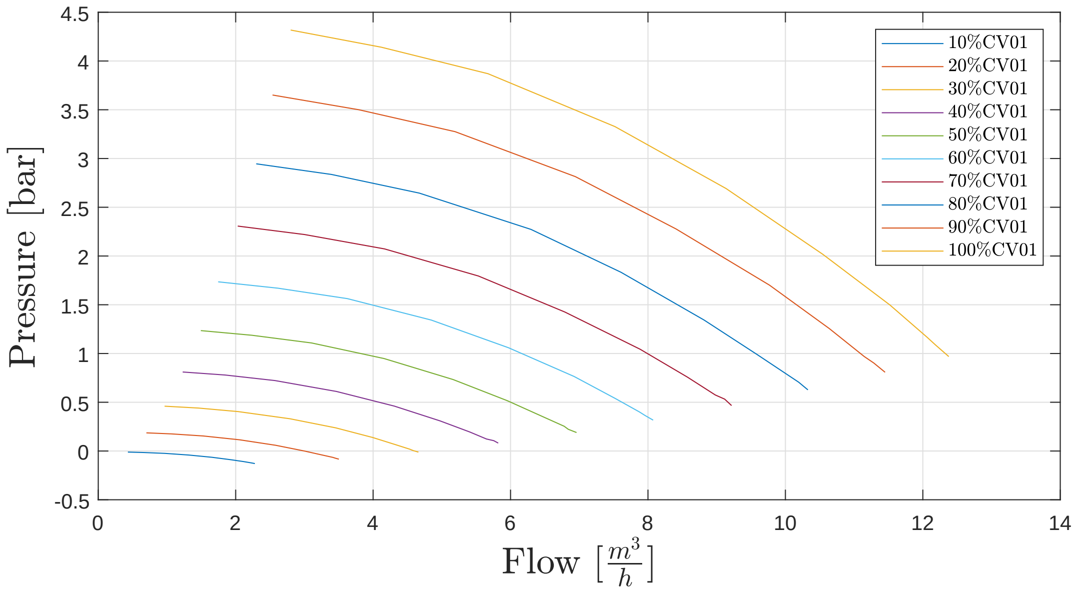

Figure 5.1: Data Points for Flow vs. Pressure

Figure 5.2 shows data points connected together, representing a pump power consumption curve.

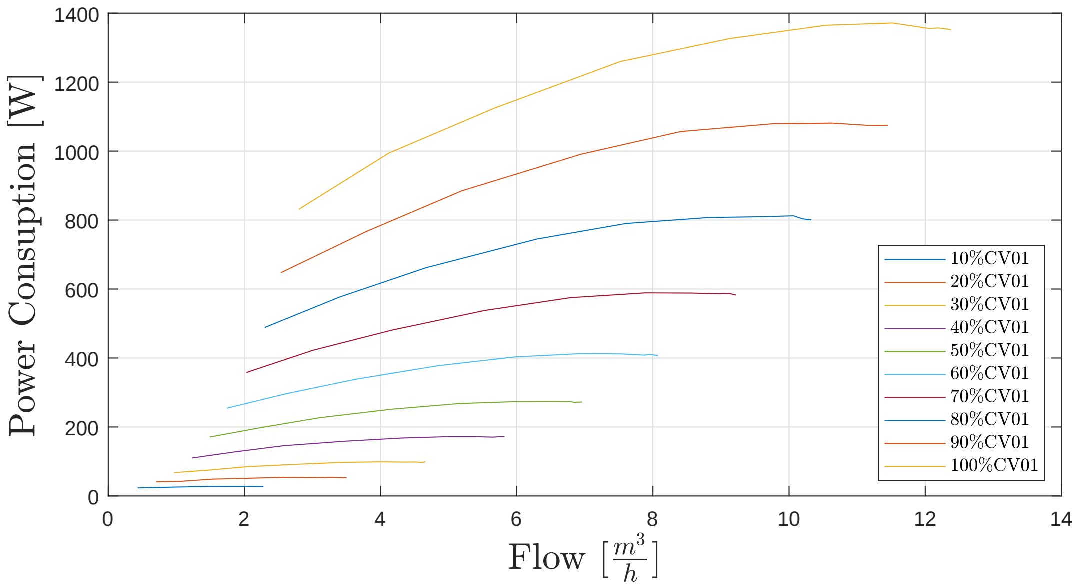

Figure 5.2: Data Points for Flow vs. Power Consumption

The ML tool *cftool* [21] was used to determine coefficients from each curve in Figures 5.1 and 5.2.

The coefficients for the model were determined at 60 % pump speed, as this promised the best overall fit through the whole range.

$$a_0 = -0.03044 \qquad a_1 = 0.07635 \qquad a_2 = 1.688$$
$$b_0 = -0.2825 \qquad b_1 = -0.7147 \qquad b_2 = 54.39 \qquad b_3 = 163.7$$

5.2 Dynamic Modelling

Giving the system a unit step input, we determined the process reaction curve of the system. This has been used in order to calculate the PID coefficients when configuring the PID, using Zielger-Nichols tuning technique. The next step was to model the curve, using equation 5.5.

$$\frac{Y(s)}{U(s)} = \frac{A \cdot e^{-st_d}}{\tau \cdot s + 1} \tag{5.5}$$

[11]

- A = final value the step response settles at
- t_d = time delay
- τ = time it takes from 0.1 of final value to 0.9 final value

After analyzing the step response on the physical setup, we have determined the values to be A = 2.221, t_d = 0.75 and τ = 2.62.

We have used ML to model the step response, and the result can be seen in Figure 5.3.

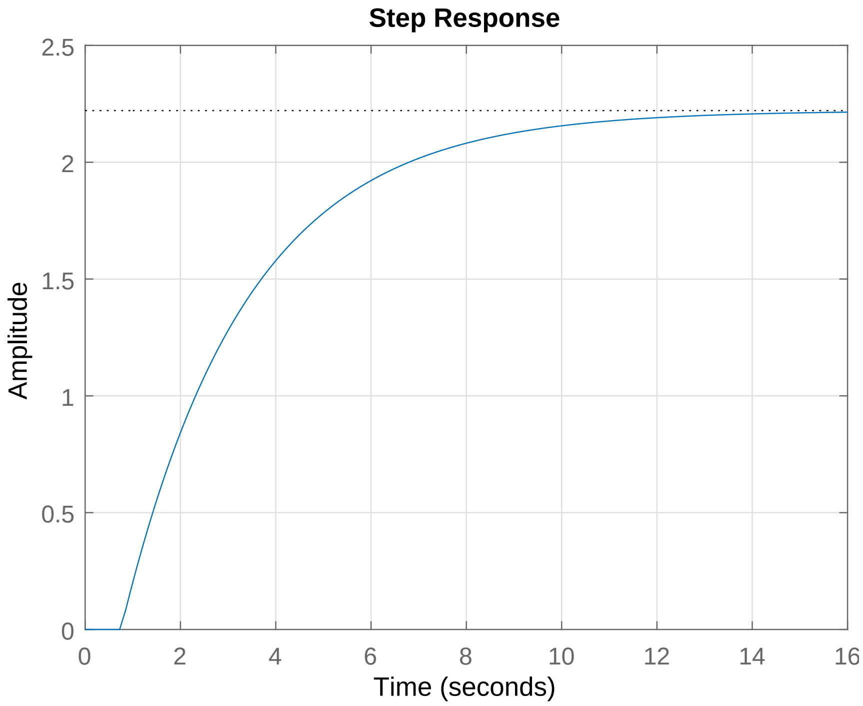

Figure 5.3: Modeled Step Response

Important to notice, that the modeled step response looks quite accurate to the actual step response. One key difference, would be the time delay. The unit step input is given instantly on the model, while on the physical setup, there was a delay of 0.5 seconds, from the moment the simulation had started, to the moment the unit step was given. The next step was to implement the Ziegler-Nichols PID tuning method. The models obtained have been accurate with some degree of error and they can be seen in Figure 5.4 and 5.5.

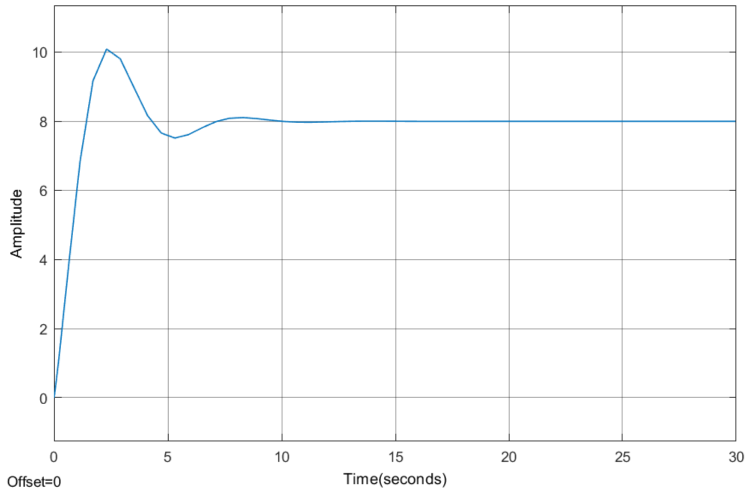

Figure 5.4: Modeled PI Controller

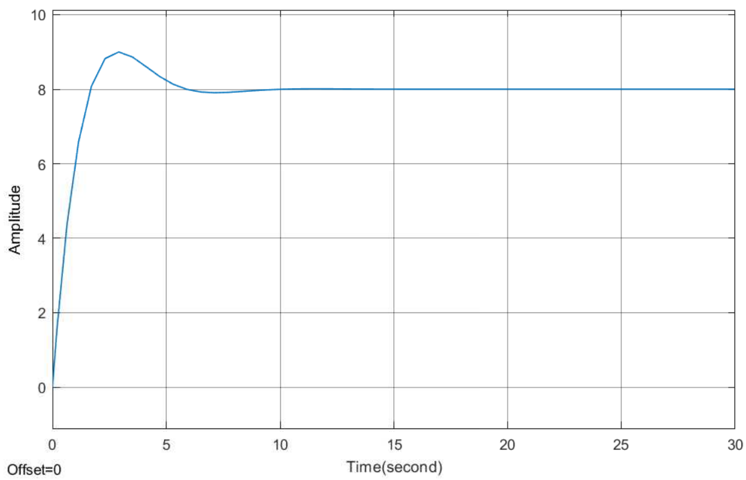

Figure 5.5: Modeled PID Controller

Both models predict with a relative high accuracy the behavior of the physical setup, with a certain degree of error. Comparisons can be seen in Chapter 6 and Appendix B.

Chapter 6

Model Validation

6.1 Static Modelling

To validate the model acquired in Chapter 5, we ran a second test with the same setup as described in Chapter 4. Coefficients were determined for pump models, for every percentage of the valve opening.

Equation 6.1 and 6.2 represent the modeled Flow and Power Consumption for 60 % pump speed.

$$H(\omega) = \frac{\bar{a_0}}{\bar{\omega_0^2}} \cdot \omega^2 + \frac{\bar{a_1}}{\bar{\omega_0}} \cdot \omega \cdot Q(\omega) + \bar{a_2} \cdot Q(\omega)^2 \tag{6.1}$$

$$P(\omega) = \frac{\bar{b_0}}{\bar{\omega_0^3}} \cdot \omega^3 + \frac{\bar{b_1}}{\bar{\omega_0^2}} \cdot \omega^2 \cdot Q(\omega) + \frac{\bar{b_2}}{\omega_0} \cdot \omega \cdot Q(\omega)^2 + b_3 \cdot Q(\omega)^3 \tag{6.2}$$

The coefficients for the model were determined at 60 % pump speed and can be seen below.

$$\begin{aligned} \bar{a_0} &= -0.03044 & \bar{a_1} &= 0.07635 & \bar{a_2} &= 1.688 & & \\ \bar{b_0} &= -0.2825 & \bar{b_1} &= -0.7147 & \bar{b_2} &= 54.39 & \bar{b_3} &= 163.7 \\ \bar{\omega_0} &= 2298rpm & & & & & & \end{aligned}$$

Figures 6.2 and 6.1, represent the Pressure and the Power Consumption for one of the tests.

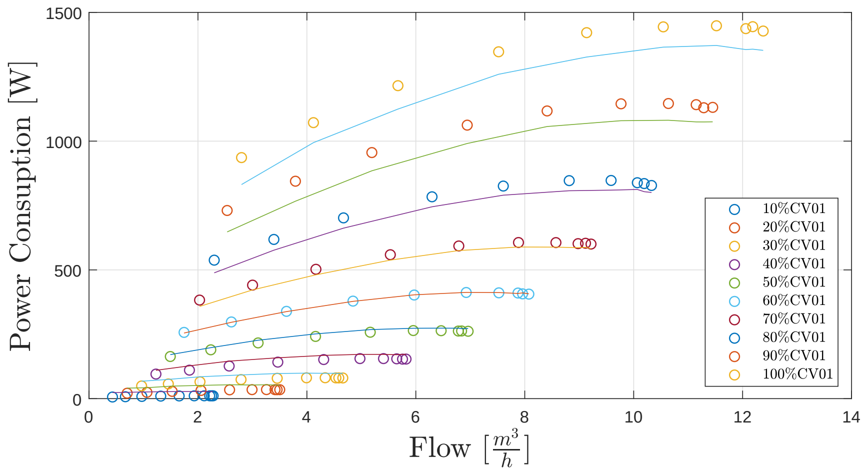

Figure 6.1: Flow Vs. Modeled Power Consumption

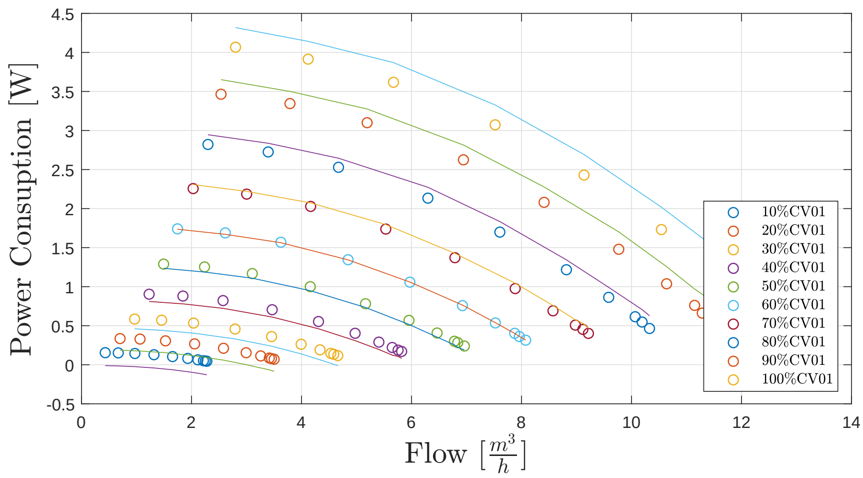

Figure 6.2: Flow Vs. Modeled Pressure

As expected, the models are able to approximate the pump curves, that are closer to the actual pump speed the models were determined for. They are not as accurate overall as expected. The modelling errors could come as a result of many factors, however, there is room for improvement. As with the case of the pressure sensors, we were not able to determine accurate values for the pump speeds. Again, we have used gains and offsets provided to us. Additionally, motor and drive efficiency are not accounted for, as done in previous research [22].

6.2 Dynamic Modelling

We determined the coefficients for our P, I, and D using Ziegler Nichols tuning method. The results both in theory and practice can be seen below. Figure 6.3 represents the model for the PI - Controller, while Figure 6.4 represents the output of the system using the same controller.

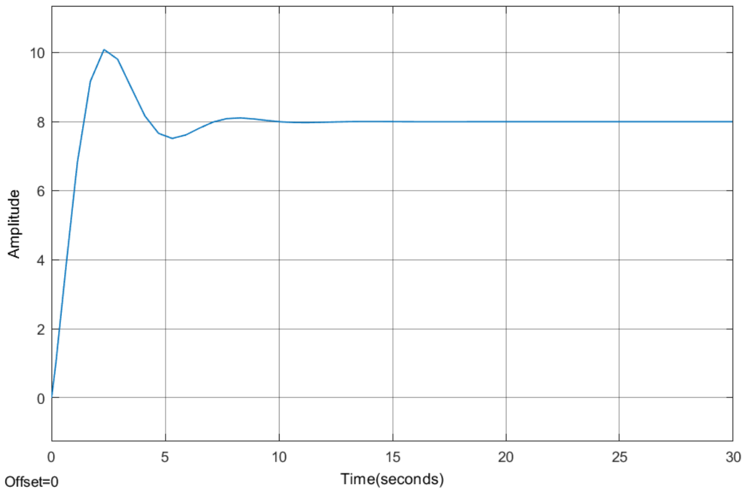

Figure 6.3: Modeled PI Controller

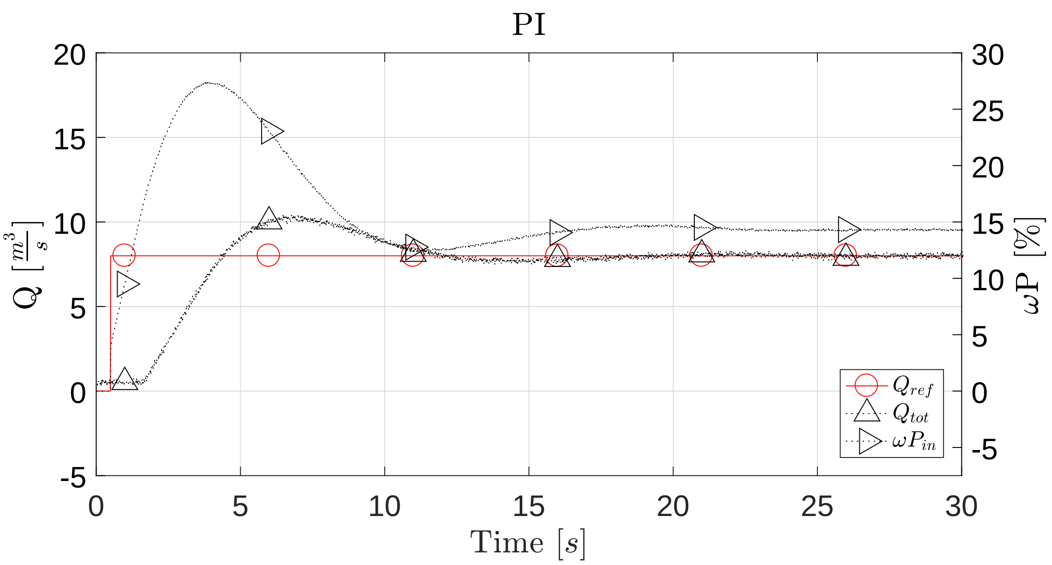

Figure 6.4: Actual PI Controller

The output of the model is quite similar to the output of the setup, with the exception of the delayed step input. Additionally, the model seems to converge faster to the setpoint.

The model for the PID - Controller also presents strong similarities with the actual output of the setup. The comparison can be seen in figures 6.5 and 6.6.

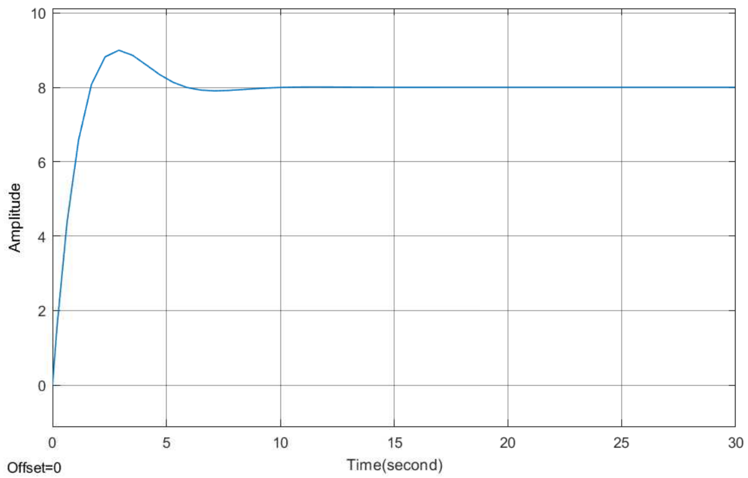

Figure 6.5: Modeled PID Controller

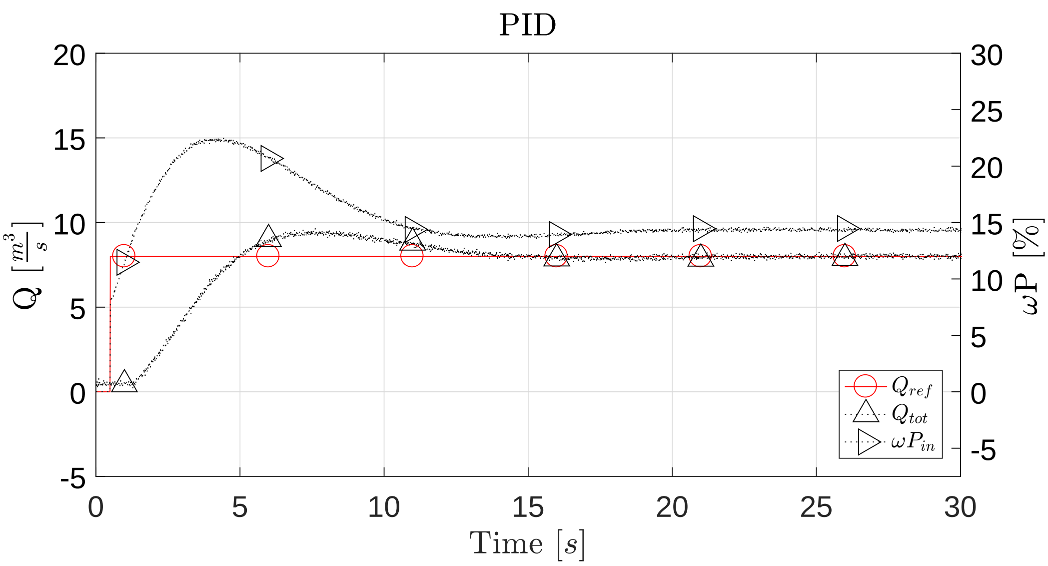

Figure 6.6: Actual PID Controller

Same can be noticed for the PID - Controller, the step input for the physical setup is given 0.5 seconds later. The model also converges faster to the setpoint. All figures can be found in Appendix C.

Chapter 7

Controller Design

There exist many different approaches to designing a controller, all of which have their advantages and disadvantages. For this project we decided to use a very simple approach, the Proportional Derivative Integral (PID) controller. This chapter compares this to another possible option, the state-space controller, explains both of them briefly and argues for our choice.

7.1 PID control

A PID controller consists of three parts, a proportional, integral and derivative part, which it inherits its name from. A mathematical representation can be given by one of the two equations in Equation 7.1.

$$\begin{aligned} D_{cl}(s) &= k_P + \frac{k_I}{s} + k_D s \\ D_{cl}(s) &= k_P(\frac{T_I}{s} + T_D s) \end{aligned} \tag{7.1}$$

[11]
These equations can be visualised in block diagrams, as shown in Figure 7.1.

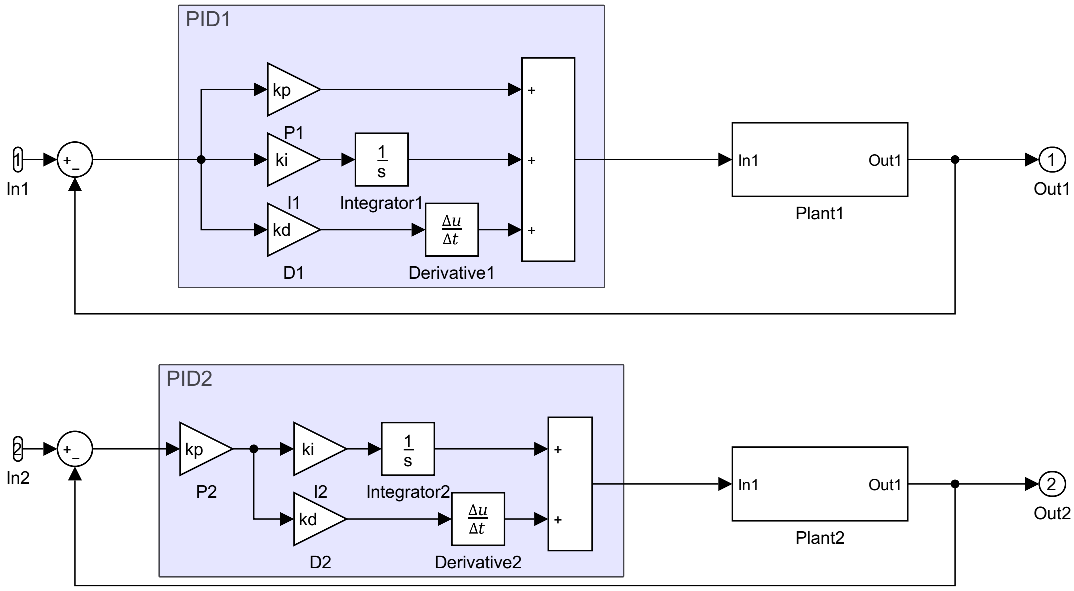

Figure 7.1: Comparison of two common PID designs

Figure 7.1 also shows the typical placement of a PID-controller, on the forward loop, behind the subtraction of the feedback and before the input to the plant.

The gains k_P, k_I and k_D have to be tuned according to the dynamics of the plant and the desired characteristics of the controller. This tuning can be done manually, or methodically. We chose to use the Ziegler-Nichols tuning method, as described by Franklin et al. [11], as this method promised results close to our requirements and was discussed in our courses.

To use this method, some knowledge about the system is required. If the system reacts stable to a step input, the output measured can be used to tune k_P, k_I and k_D according to the Quarter Decay Method (QDM) [11].

In the case that the system has an unstable reaction, the Ultimate Sensitivity Method (USM) should be used. Here the k_p is increased until the system becomes marginally stable. [11]

In both cases the output of the system is to be graphed over time and from the corresponding graph some characteristic values can be read. Because we are using the quarter decay method, we won't discuss the ultimate sensitivity method any further.

As mentioned before, to use the QDM, we need a step response of the system. The following section explains step response in more detail and shows our analysis.

7.1.1 Quarter Decay Tuning

To obtain all needed data, the step response of the system needs to be analysed.

Step Response

The step response needs to be measured on the Open Loop (OL) system. A block diagram of an OL setup can be seen in Figure 7.2.

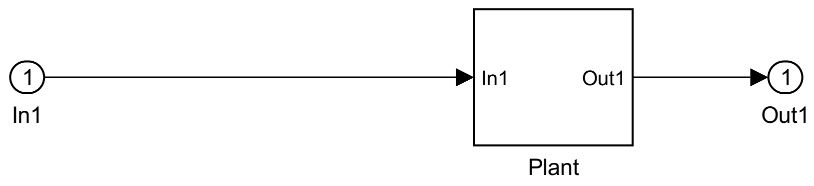

Figure 7.2: OL block diagram of a system

The recorded output can be seen in Figure 7.3. The figure also shows two red lines, approximating the slope and the final value.

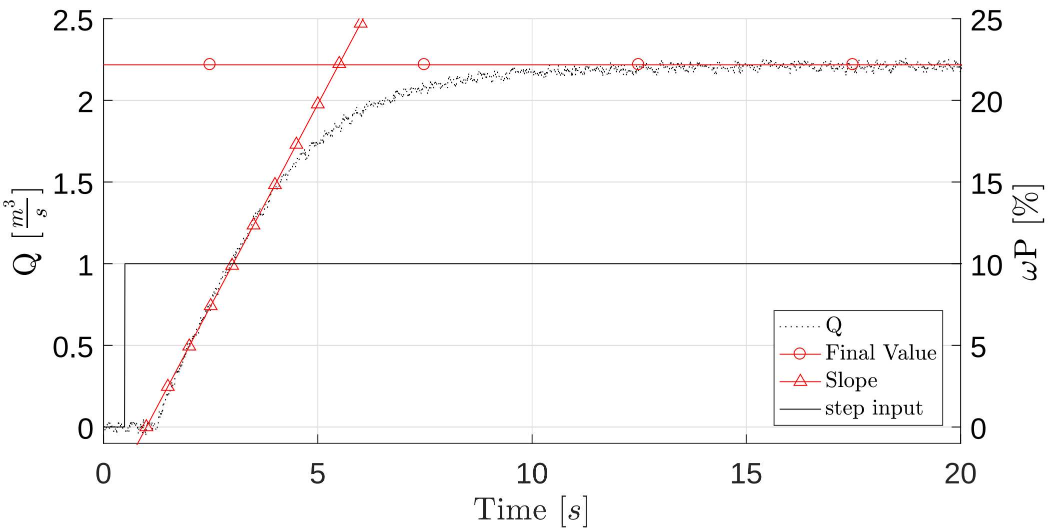

Figure 7.3: response to a step input with value 10

From this graph we can approximate the slope R and the Lag $L = t_d$. Because we are scaling the ωP down by a factor of 10, so we can directly input a percentage, we had to scale the aforementioned unit step up by a factor of 10, in order to get usable results. While encountering this problem, we also noticed, that the pumps don't spin at $\omega \leq 9\%$. When using the corrected step input, we got the measurements shown in Figure 7.3.

Our analysis of figure 7.3 gives us the following values:

$$\begin{aligned} R &= 2.0236 \quad \textit{slope} \\ t_d = L &= 0.75 \quad \textit{lag} \end{aligned}$$

Tuning

The QDM is tuning a P, PI or PID controller $D_c(s)$ with the formula shown in the second Equation 7.1. The values of k_P, T_I and T_D are scalar gains, tuned according to the characteristics obtained from figure 7.3.

Type of Controller	Optimum Gain
P	$k_P = 1/RL$
PI	$k_P = 0.9/RL$ $T_I = L/0.3$
PID	$k_P = 1.2/RL$ $T_I = 2L$ $T_D = 0.5L$

Note that the parameters T_I and T_D are not mentioned for the first two controller types, as they are to be set to 0.

Results

We tested all three options of P, PI and PID control with the tuned parameters, the results can be seen in Figure 7.4. Please note, that the P-controller with the calculated value did not achieve any results, as the error was too small to start the pump. To overcome this issue, we multiplied k_P with 10. This was solely done for the P-controller, because the PI- and PID-controller gave proper results without this correction. We expect this to be caused by the scaling factor before the input to the pumps, as discussed in Subsection 7.1.1.

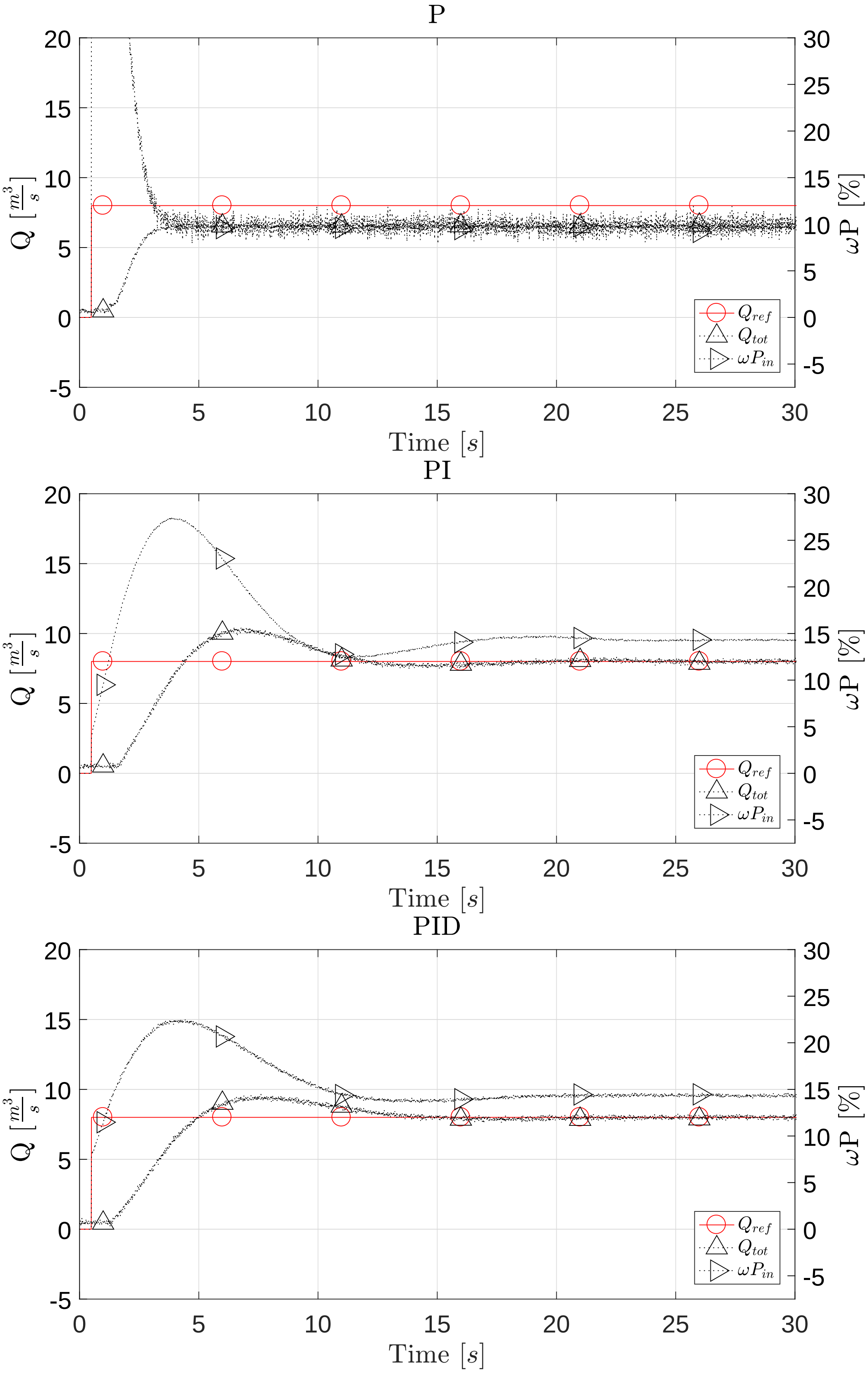

Figure 7.4: Results from testing a P, PI and PID controller with ZN tuning

As was expected, the P-controller settles the fastest, at approximately 4 seconds, while the PI and PID settle around 20 seconds. The P-controller has a very big steady-state error though, which doesn't fulfil our requirements. The PI-controller does not have a steady-state error, but the overshoot of approximately 25% doesn't fulfil our requirements either. Interestingly, the PID-controller also has an overshoot above our limits, which we expect to be caused by the ZN tuning. In the next step we tried manually tuning the k_D coefficient, to get an overshoot inside our requirements.

7.1.2 Manual Tuning

Building on top of the best candidate, the PID-controller, we manually tuned the coefficients. We came to the conclusion, that the overshoot is happening, because the controller is too aggressive, which is why we decided to decrease the k_P coefficient to 0.2. Looking back at the second formula in Equation 7.1, we can see, that this will not affect the relations between the coefficients.

Results

The results of the manual tuning can be seen in Figure 7.5.

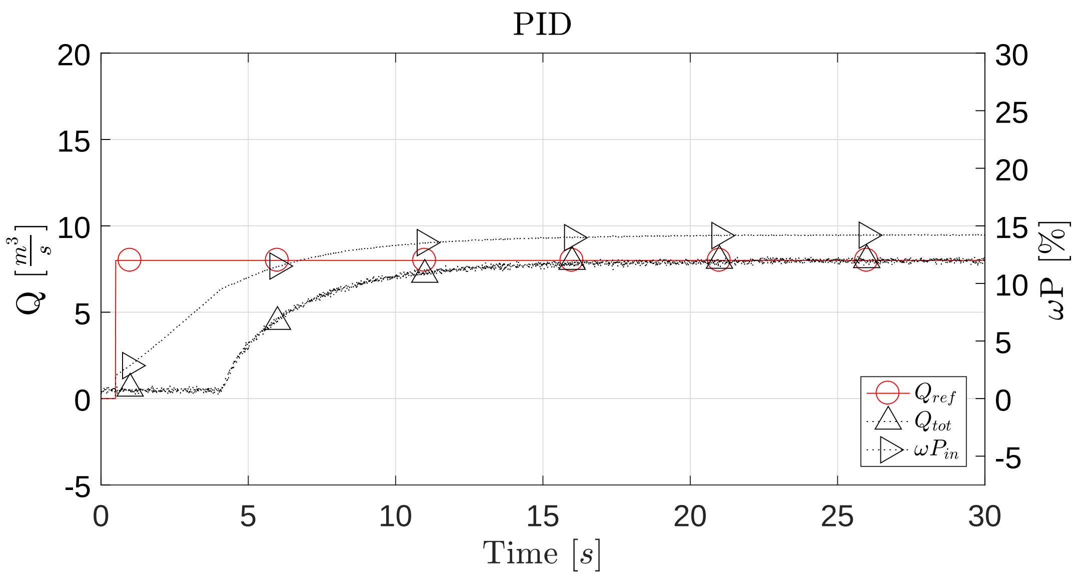

Figure 7.5: Results from ZN PID with $k_P = 0.2$

Interesting to notice here is, that the controller has a deadtime of approximately 4 seconds, before its output affects the system.

One big disadvantage of PID tuning for this system is, that it is not taking advantage of the existence of three pumps as individual inputs. The tuning in this chapter is based on using only a single pump, but could easily be repeated for multiple pumps with equal input speed. Based on research done by Pedersen and Yang [10], it seems that using multiple pumps could increase efficiency, but it be most efficient to always spin all used pumps equally. This is not possible with a single PID controller, as it only outputs one speed to all connected pumps. It could be possible to implement an additional logic, switching between differently tuned controllers for different flow requirements.

Chapter 8

Discussion

In this chapter we discuss the initially stated requirements and compare them with the results achieved. We will also have a look at future work opportunities, to be done on top of this project.
This chapter will also explain the process of working on this project.

We initially set out three primary goals for this project:

- Creation of a dynamic model for one pump
- Design of a PID controller for the flow
- Tuning of said PID controller

And three secondary goals:

- Creation of a static model for one pump
- Creation of a static model for multiple pumps
- Design of a controller for the flow, taking efficiency into account

Looking back at the project, we can now say that all primary goals and some of the secondary goals were met.
We created a dynamic model for a single pump through analysis of experimental data and assuming a transfer function of the form $\frac{Y(s)}{U(s)} = \frac{Ae^{st_d}}{\tau s+1}$. Where all coefficients were found analysing a step response. This model gave a good fit to our step response data, but didn't perform perfectly to test the actual controller.

The design and tuning of the PID controller was primarily done on the physical setup, instead of the simulation, because it was readily accessible and provided good results. This is also the reason for the dynamic model not being our first priority when it came to time management.

We initially set out to describe the whole system, with all three pumps as a MIMO system and be able to control the total flow with minimal power consumption. This was to be achieved by using multiple pumps and benefiting from the shifted maximum efficiency point as stated in previous research. When we realised that we would not be able to finish that project in the given timeframe, we chose to implement a PID control on a SISO system instead. The knowledge gained about the system from developing the static model was helpful to determine an operating range for the PID control. It showed that almost identical dynamics were to be expected at all points $0\% < CV_1 \leq 100\%$ and $10\% \leq \omega P \leq 100\%$. It also evens the path for future work on energy efficiency, because there already exists a reliable model of the power consumption with respect to H and Q, which can easily be extended to a model of the efficiency.

- Maximum Overshoot $M_p = 0\%$
- Steady-state error $e_{ss} \leq 1\%$

With the manually tuned PID controller, all requirements set in 2.3.1 and shown above were hit, but this was done by making the system very slow and creating a very big delay in reaction.

With more intensive tuning and more reasonable requirements, better coefficients for a PID controller could possibly be found. Specifically requiring no overshoot is not expected to be reasonable in most applications of a flow controller. It might for example be more beneficial to require the integral of the error to be very small, to ensure steady flow on average.

8.1 Future Work

As with every project, not all work on this topic is done yet. We therefore propose a small list of future work opportunities. Some of the points on that list are inspired by our initial goal, energy efficiency.

Development of a more robust dynamic model. As stated in Section 5.2, our dynamic model was not a perfect description of the system. With more advanced modelling techniques and more research into this, a better model could be found. This would benefit most other future work.

While developing our static model, we found that it was not accurate for the whole operating range of the system. We believe it is possible to find a model better suited, if more work is put into this topic. This would also benefit most other future work and maybe also the general understanding of pumping systems.

Research in the industry to find what requirements actually matter should be conducted, to ensure that the next iteration of this controller would be useful.

Based on our research alone, we cannot know which factors to prioritise and therefore not build a beneficial controller.

Our initial goal of efficiency optimisation was not met, due to a shift in focus. We still think this is a worthwhile goal to work on. Since we already put some thought into the topic, we propose building a decisive logic to decide how many pumps to use, which might be possible to implement as a lookup table. A more reliable approach would of course be modelling the efficiency of 1, 2 and 3 pumps according to Q and H and developing a MIMO controller taking those factors into account. Developing a reliable model for the efficiency could help building a lookup table or decisive logic to switch between multiple pumps.

Chapter 9

Conclusion

From the Problem Delimitation in Chapter 2 there were three primary goals for this project.

A dynamic model was created according to a very simple transfer function. All variables were gathered through a step response test. This helped achieve a dynamic model very similar to the system response throughout the whole operating range.

The development of a static model through polynomial fitting helped understand the pumps capabilities and set a reasonable setpoint for controller testing. It also made it possible to get an idea of the power consumption and will help in future work towards efficiency optimisation.

All stated requirements were hit, even though we conclude they might not fit to real world requirements.

As stated in Section 8.1 possible future work could be aiming towards energy efficiency. Some steps required in that direction would be better modelling and more complex control. The final goal for a better controller could be maintaining a set flow at maximal efficiency, by using optimal scheduling of multiple pumps in parallel.

Bibliography

[1] Daniel Busemann, Troels Klein, and Razvan Bucur. *P4—Control-of-a-Three-Pump-System - (a 4th Semester Project)*. https://github.com/BlackAndWhiteSnaable/P4---Control-of-a-Three-Pump-System. 2018. URL: https://github.com/BlackAndWhiteSnaable/P4---Control-of-a-Three-Pump-System.

[2] Howard Perlman and USGS. *The water in you*. Accessed May. 9, 2018. Dec. 2016. URL: https://water.usgs.gov/edu/propertyyou.html.

[3] One Medical. *Do I really need to drink 8 glasses of water a day?* Accessed May. 9, 2018. Aug. 2017. URL: https://www.onemedical.com/blog/live-well/daily-water-intake/.

[4] *Surface Flow Control*. Accessed May. 9, 2018. 2015. URL: https://www.geoilandgas.com/oilfield/surface-flow-control.

[5] *'Ultimate Flow Control' at the Brew House*. http://www.valveuser.com/1809-ultimate-flow-control-at-the-brew-house.htm. Accessed May. 9, 2018. 2016. URL: http://www.valveuser.com/1809-ultimate-flow-control-at-the-brew-house.htm.

[6] Accessed May. 9, 2018. URL: https://www.ssdairyequipments.com/flow-control-valve.html.

[7] *Flow Measurement & Control*. Accessed May. 9, 2018. URL: https://mena-water.com/products/flow-measurement-control/.

[8] Investopedia Staff. *DEFINITION of 'Lifetime Cost'*. Accessed May. 9, 2018. May 2007. URL: https://www.investopedia.com/terms/l/lifetime_cost.asp.

[9] Michael W. Volk. *Pump characteristics and applications*. 3rd ed. CRC Press, 2014. Chap. 6, pp. 326–335, 103.

[10] Gerulf K.M. Pedersen and Zhenyu Yang. "Efficiency Optimization of a Multi-pump Booster System". In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. GECCO '08. New York, NY, USA: ACM, 2008, pp. 1611–1618. ISBN: 978-1-60558-130-9. DOI: 10.1145/1389095.1389400. URL: http://doi.acm.org/10.1145/1389095.1389400.

[11] Gene F. Franklin, J. Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems, Global Edition*. Pearson Education Limited, Aug. 5, 2014. 880 pp. ISBN: 978-1292068909. URL: https://www.ebook.de/de/product/24639577/gene_f_franklin_j_powell_abbas_emami_naeini_feedback_control_of_dynamic_systems_global_edition.html.

[12] *Pump Handbook*. https://www.grundfos.com/campaigns/download-the-pump-handbook.html. Grundfos, 2004.

[13] *The Centrifugal Pump*. Grundfos. URL: http://machining.grundfos.com/media/16620/the_centrifugal_pump.pdf.

[14] Larry Bachus and Angel Custodio. *Know and understand centrifugal pumps*. Elsevier, 2003.

[15] Grundfos. URL: https://product-selection.grundfos.com/product-detail.product-detail.html?from_suid=152744921238808108966228431007&pumpsystemid=381660835&qcid=381661654.

[16] *Target Computer Setup*. URL: https://se.mathworks.com/help/xpc/configuration-basics-1.html.

[17] *Rosemount 8732E Field Mount Magnetic Flow Meter Transmitters*. URL: http://www.emerson.com/en-us/catalog/rosemount-8732e.

[18] ZES ZIMMER Electronic Systems GmbH. *LMG670*. URL: https://www.zes.com/en/Products/Precision-Power-Analyzers/LMG670.

[19] *About Rosemount 3051S Series of Pressure Instrumentation*. URL: http://www.emerson.com/en-us/automation/measurement-instrumentation/pressure-measurement/pressure-transmitters-and-transducers/about-rosemount-3051s-series-of-instrumentation.

[20] Kasper L Jepsen et al. "Power consumption optimization for multiple parallel centrifugal pumps". In: *Control Technology and Applications (CCTA), 2017 IEEE Conference on*. IEEE. 2017, pp. 806–811.

[21] MATLAB. *Curve Fitting Toolbox*. https://se.mathworks.com/help/curvefit/index.html. Accessed May. 9, 2018. 2018. URL: https://se.mathworks.com/help/curvefit/index.html.

[22] Zhenyu Yang and Hakon Børsting. "Optimal scheduling and control of a multi-pump boosting system". In: *Control Applications (CCA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 2071–2076.

Appendix A

System Overview

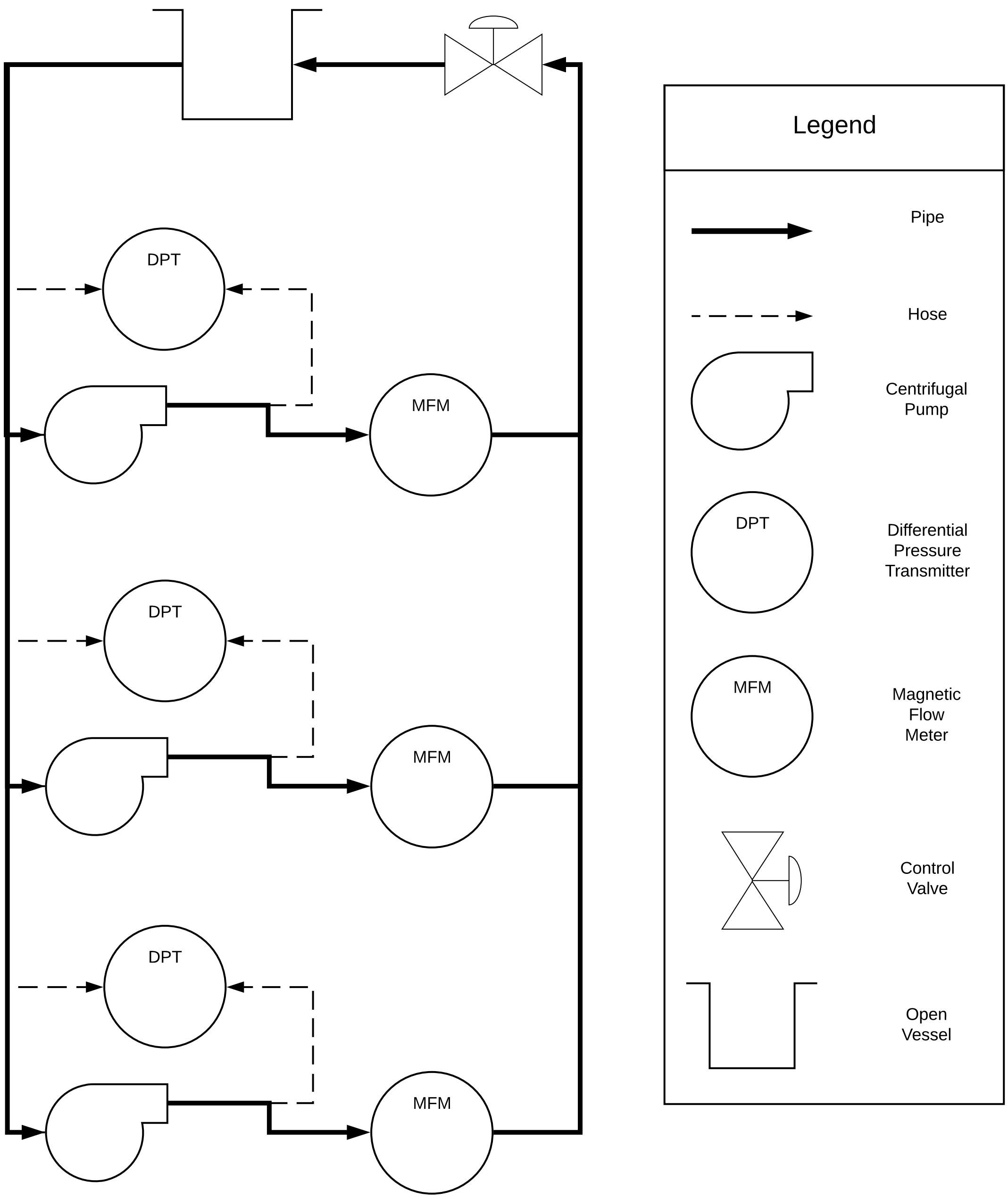

Figure A.1: Schematic Overview of the System

Appendix B

Modelling

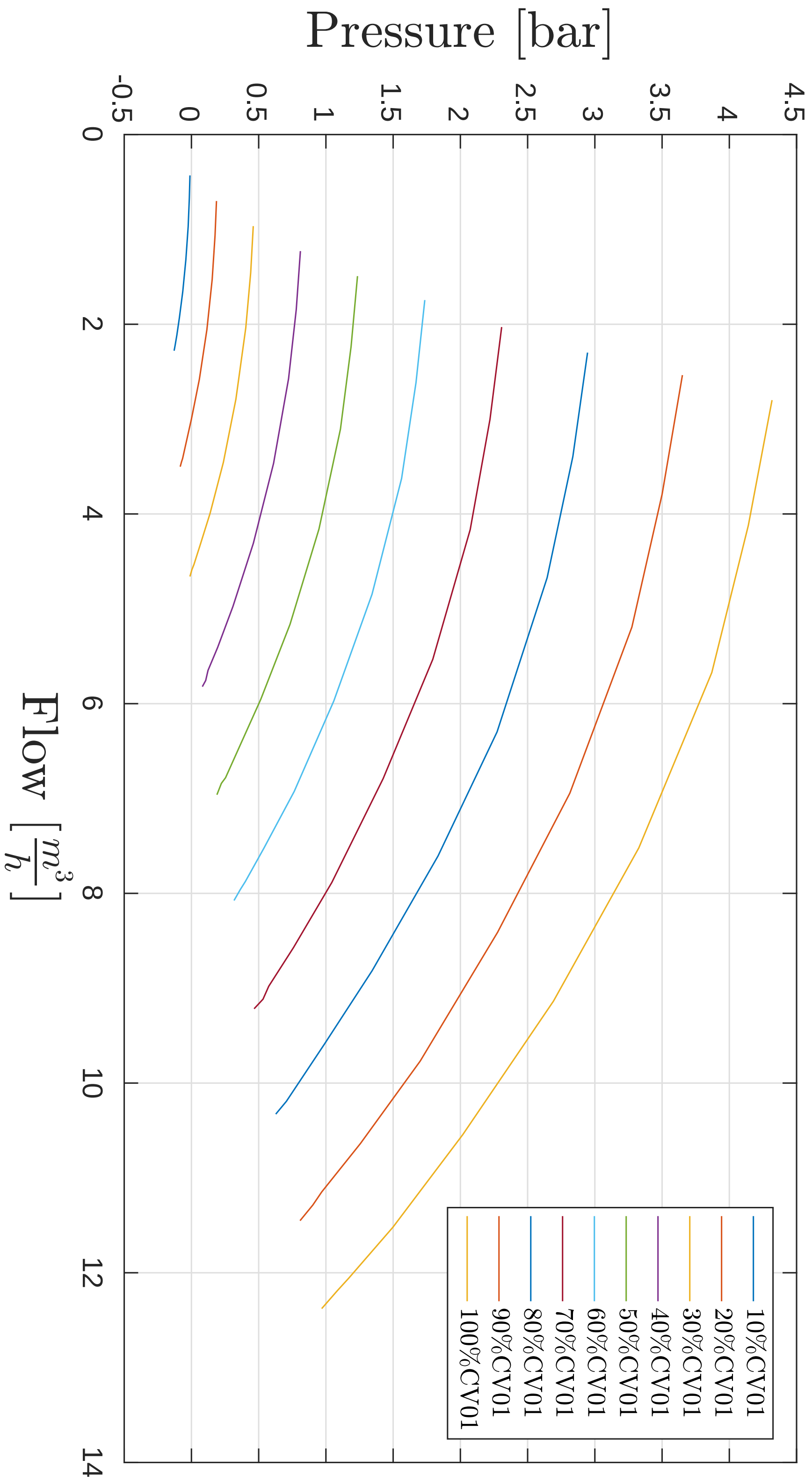

Figure B.1: Data Points for Flow vs. Pressure

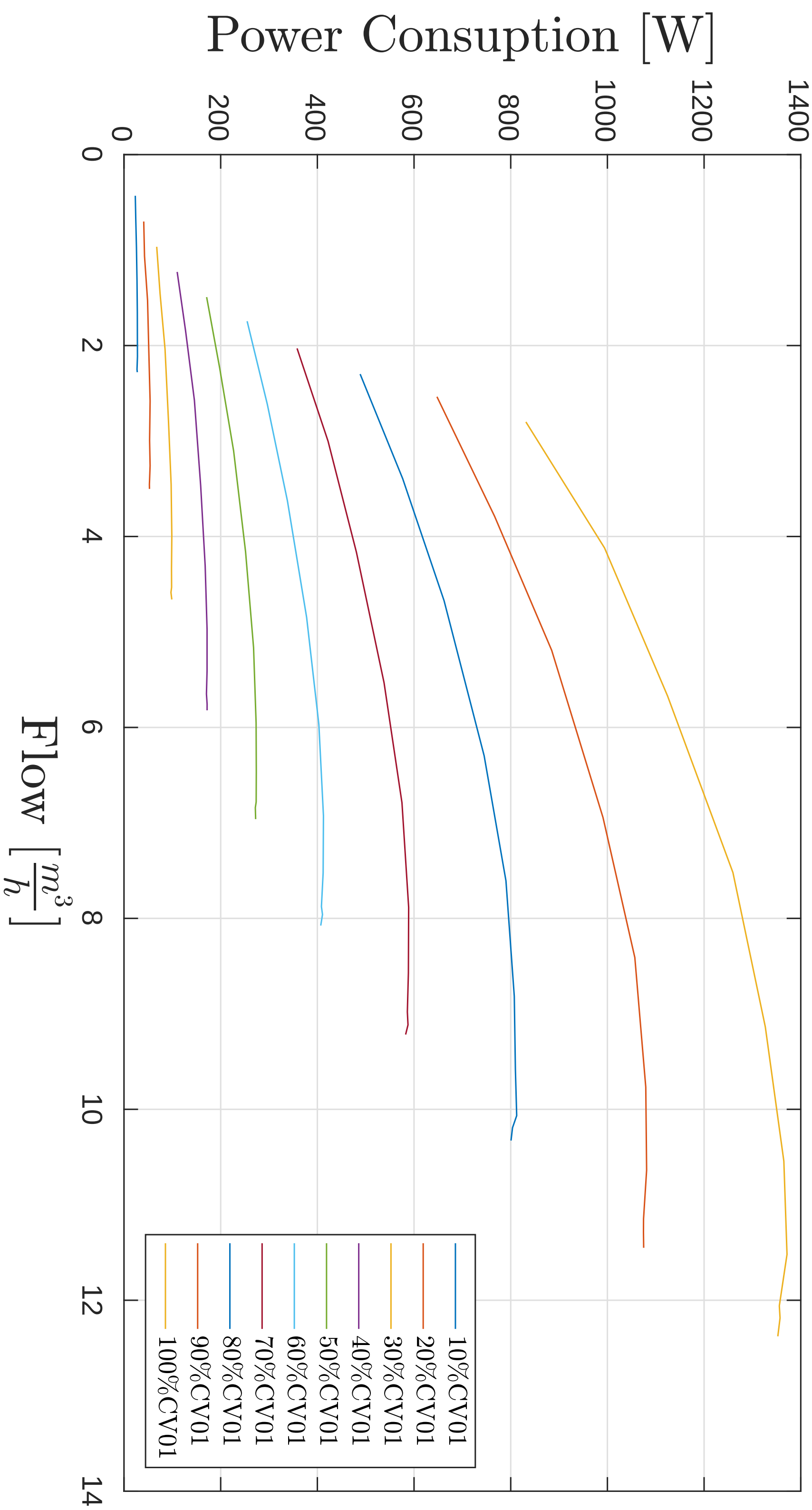

Figure B.2: Data Points for Flow vs. Power Consumption

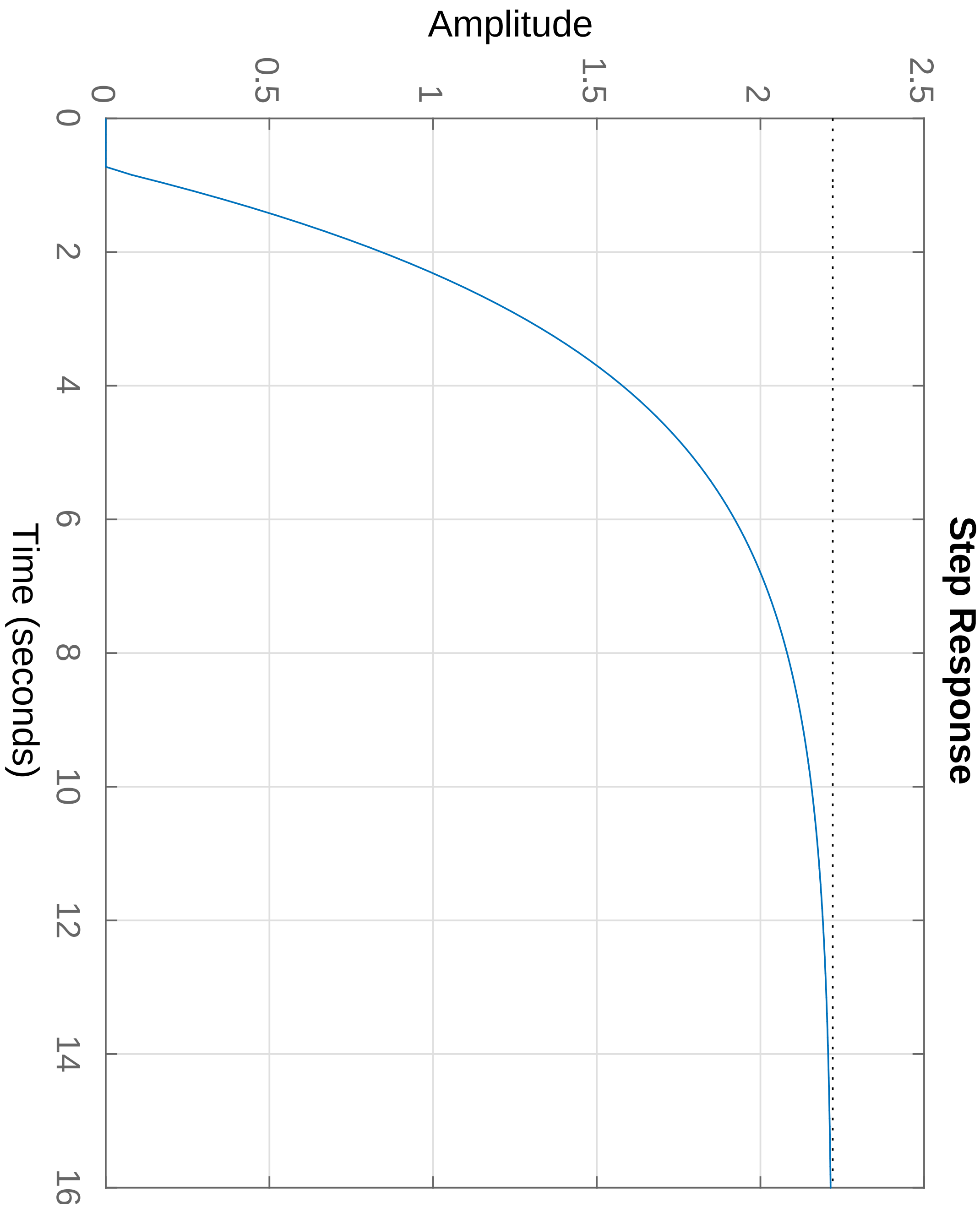

Figure B.3: Modeled Step Response

Appendix C

Model Validation

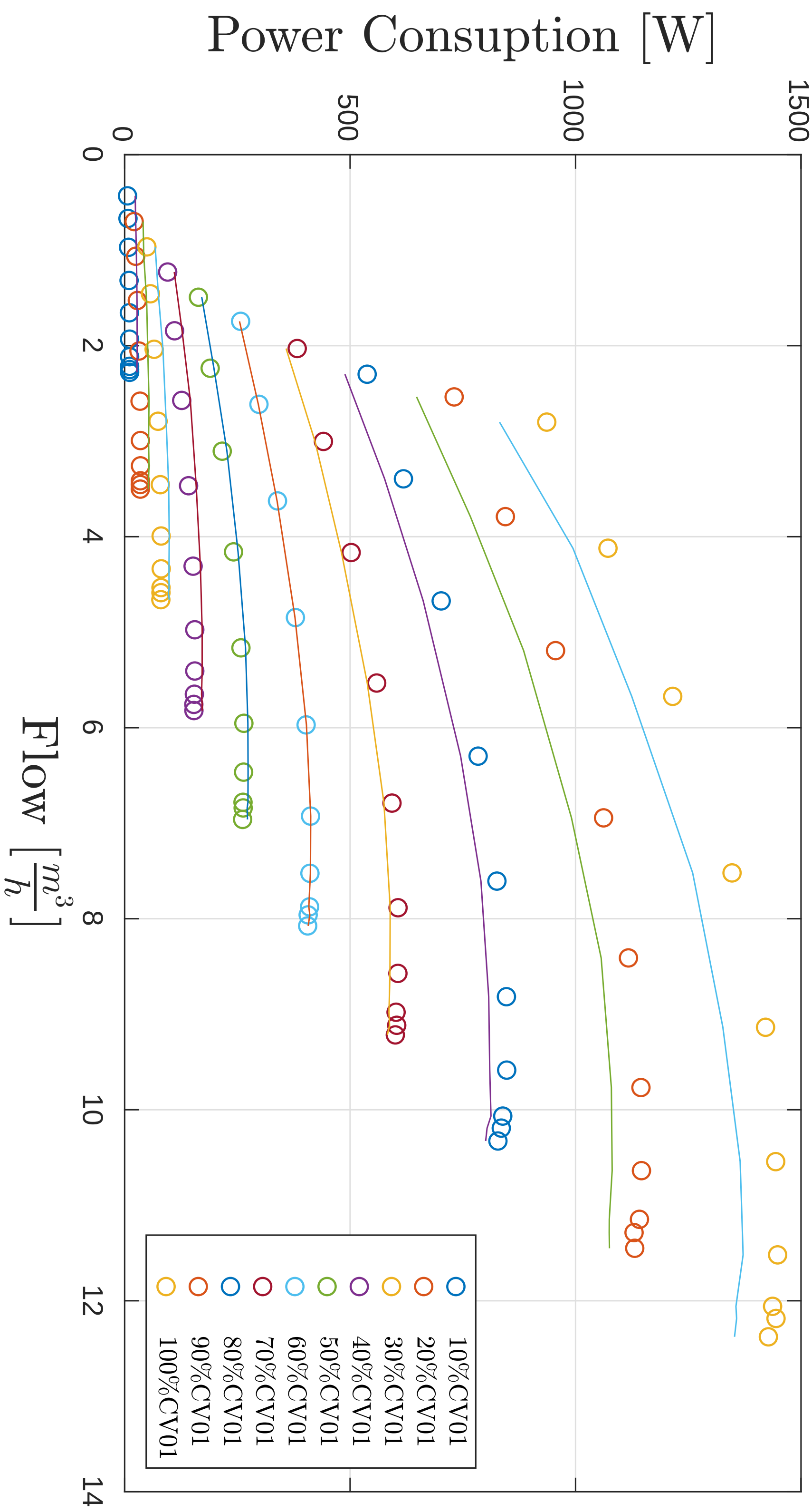

Figure C.1: Flow Vs. Modeled Power Consumption

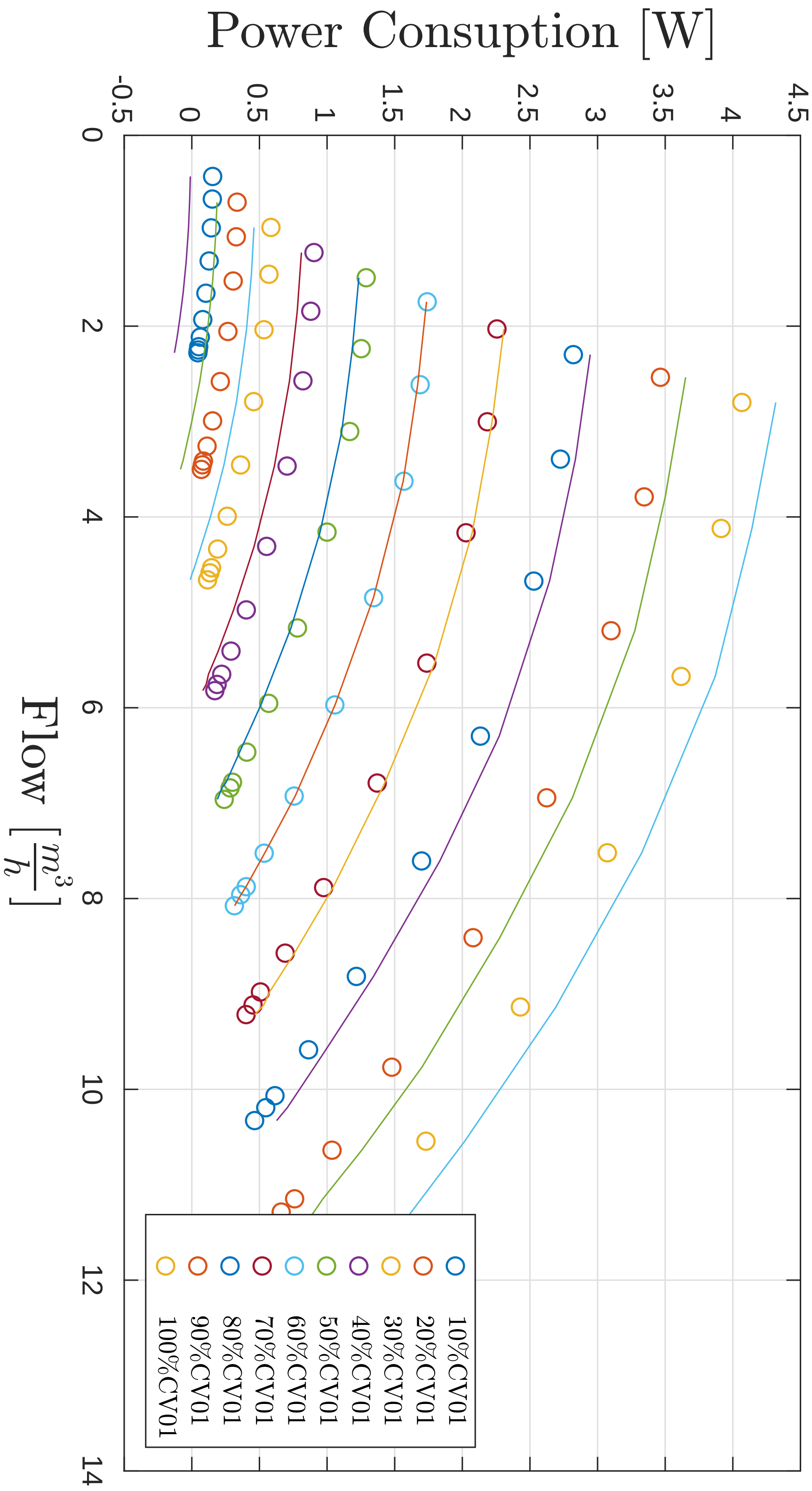

Figure C.2: Flow Vs. Modeled Pressure

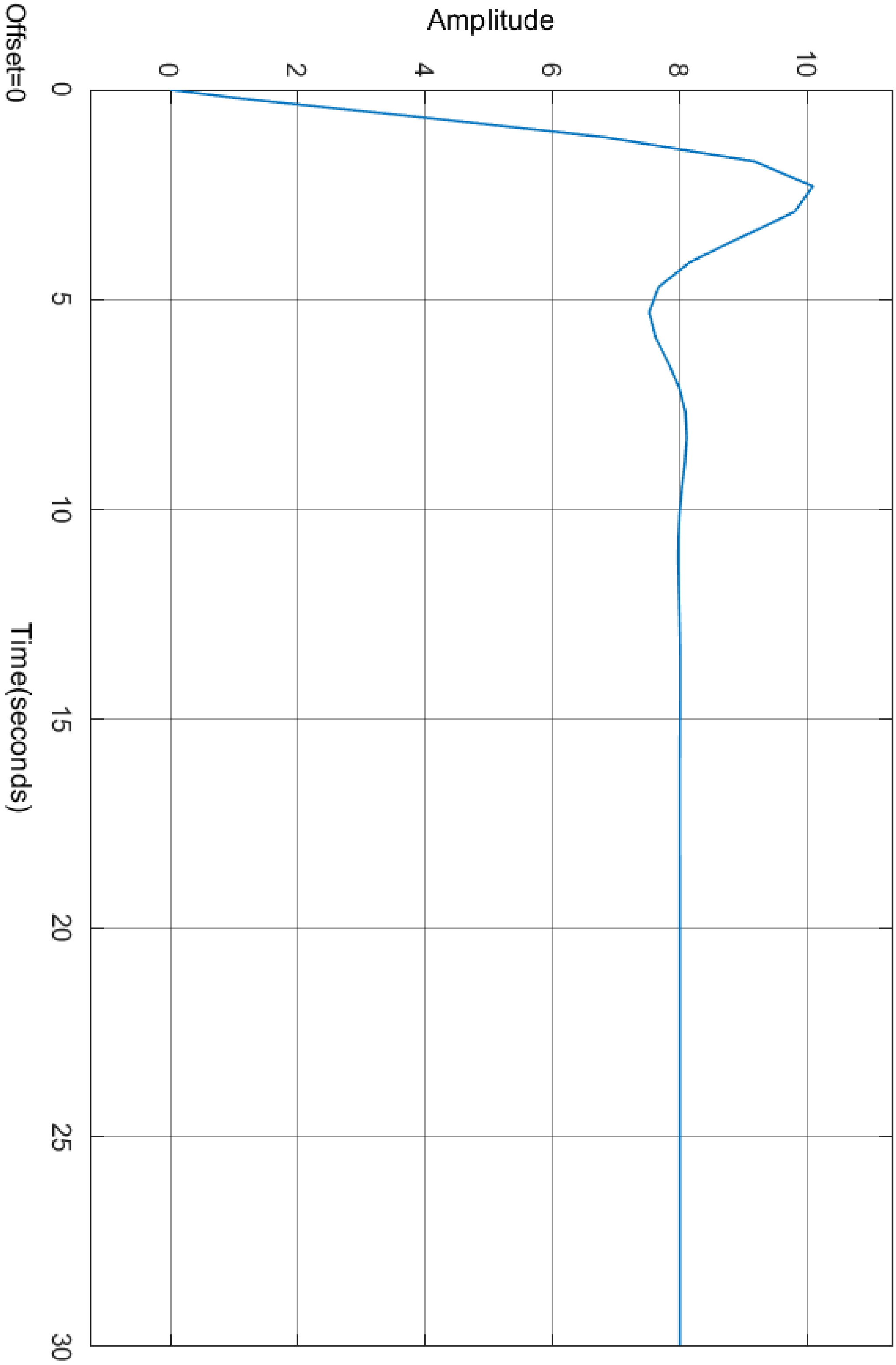

Figure C.3: Modeled PI Controller

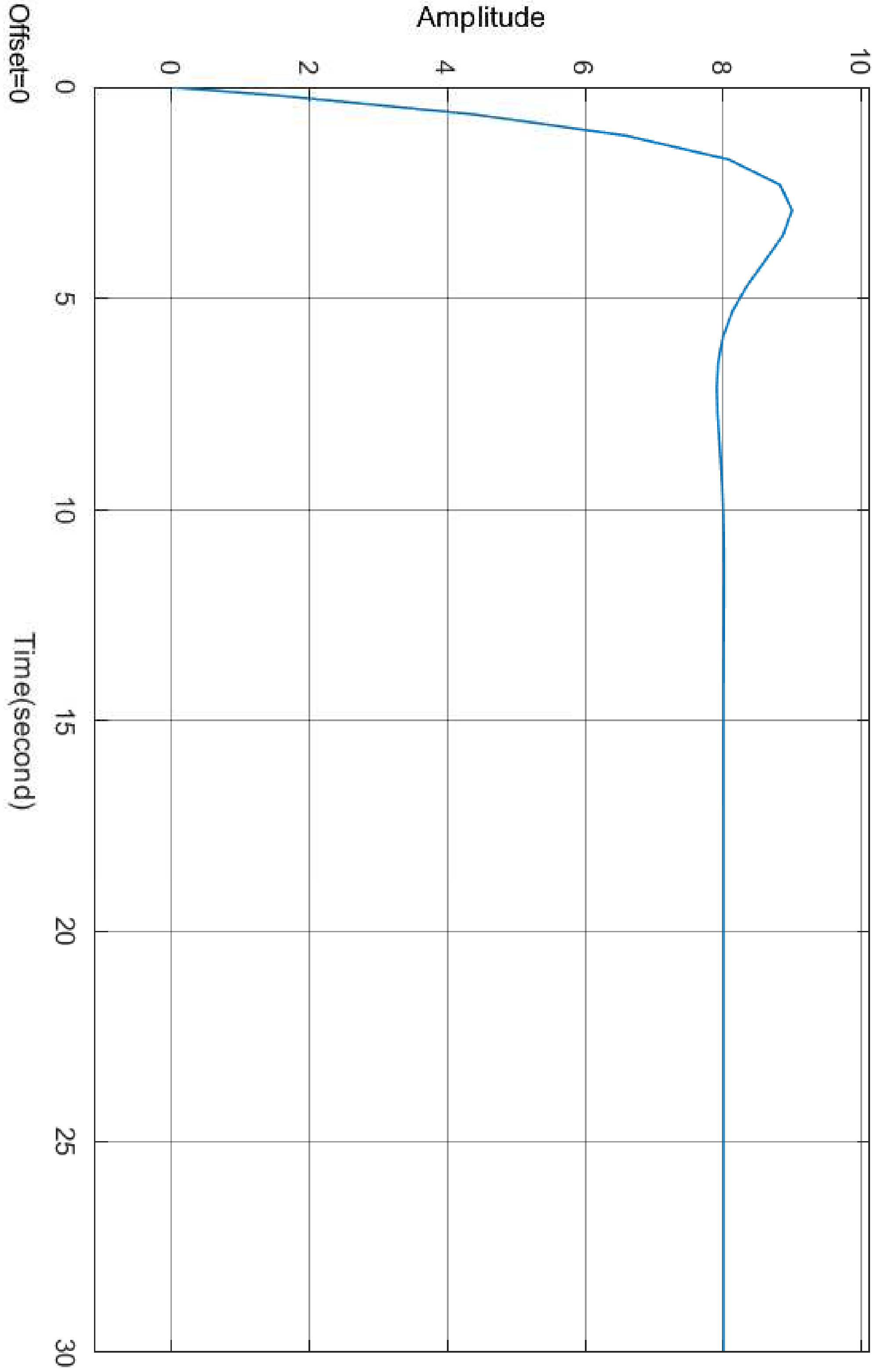

Figure C.4: Modeled PID Controller

Appendix D

Measured Data

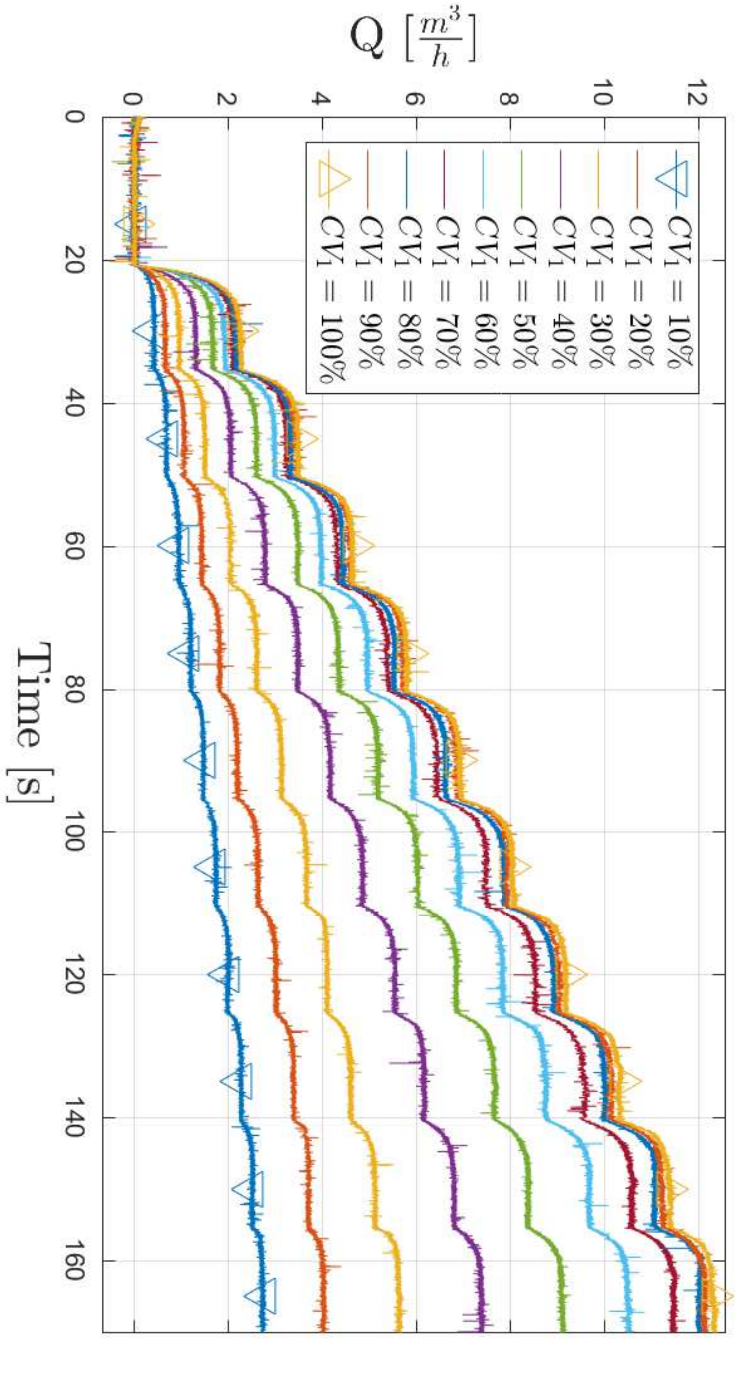

Figure D.1: Measured Flow

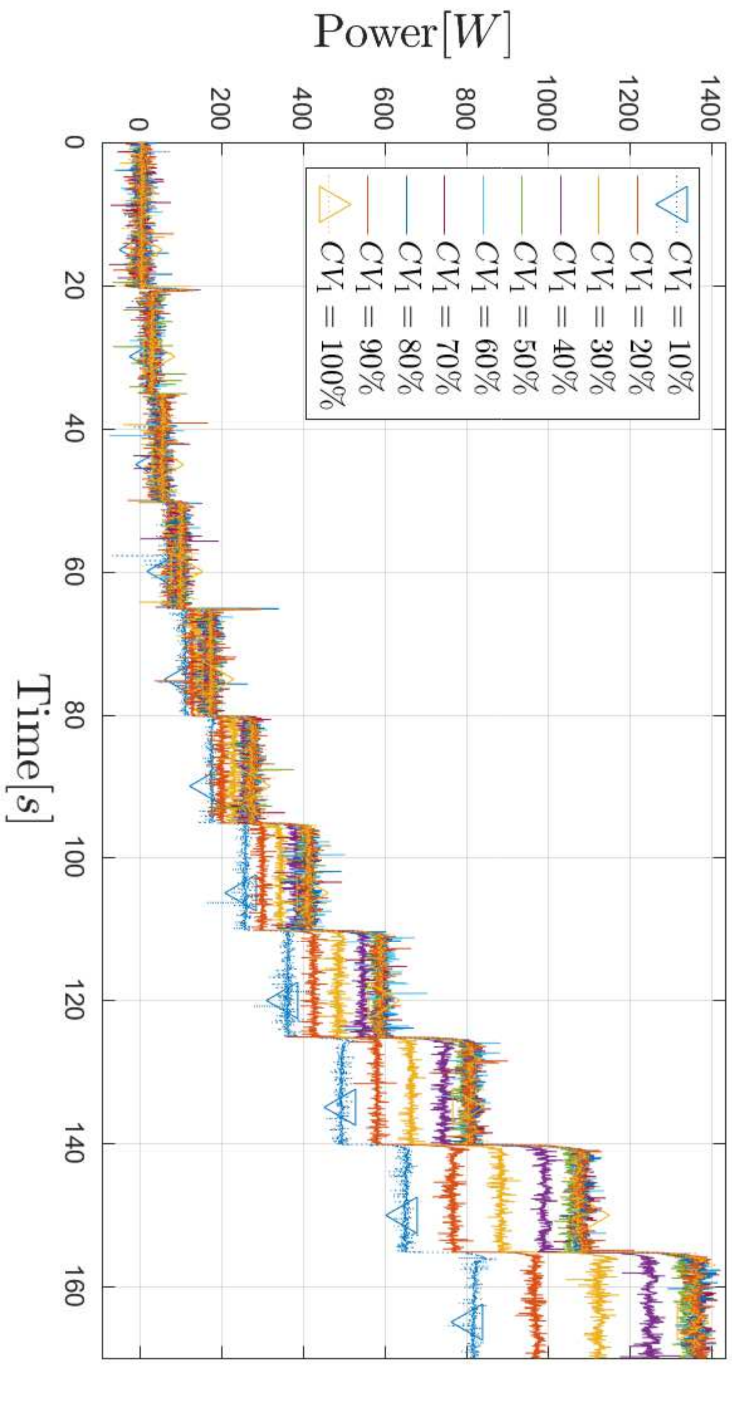

Figure D.2: Measured Power

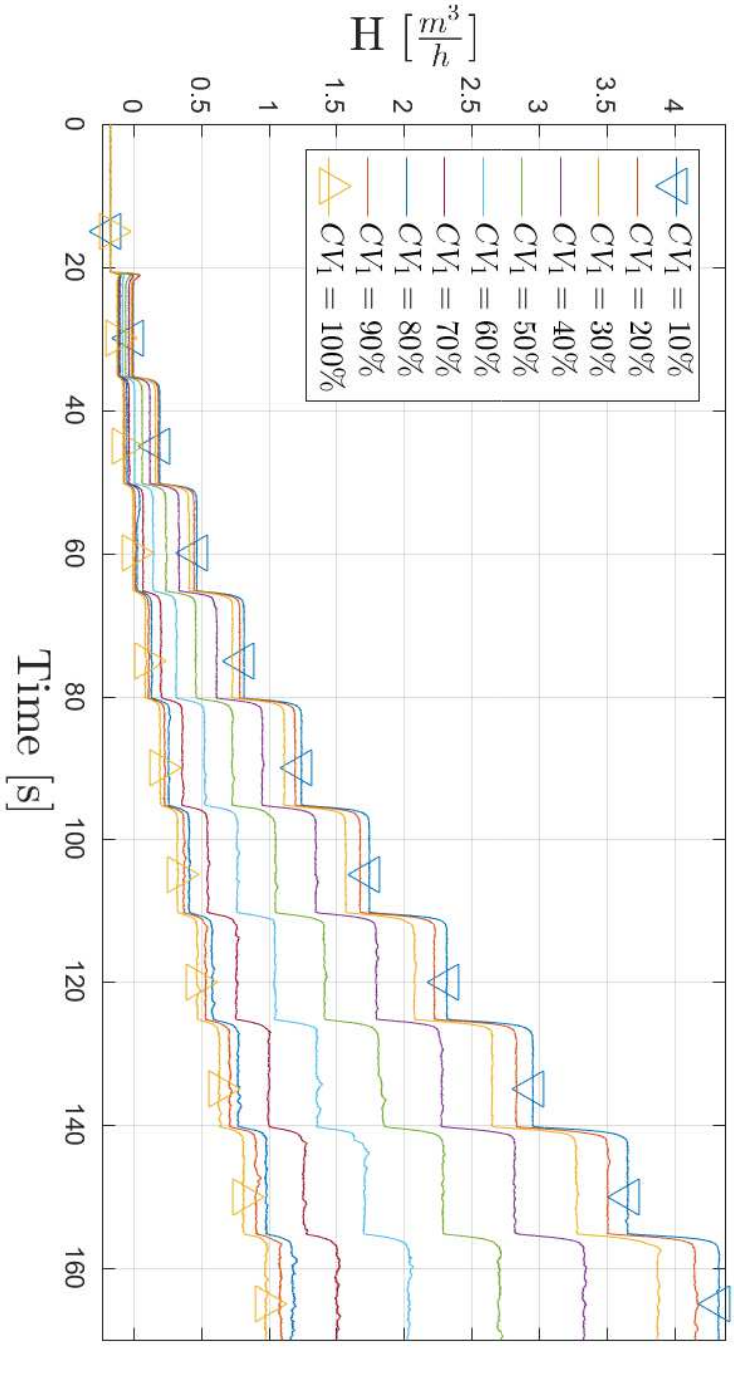

Figure D.3: Measured Pressure

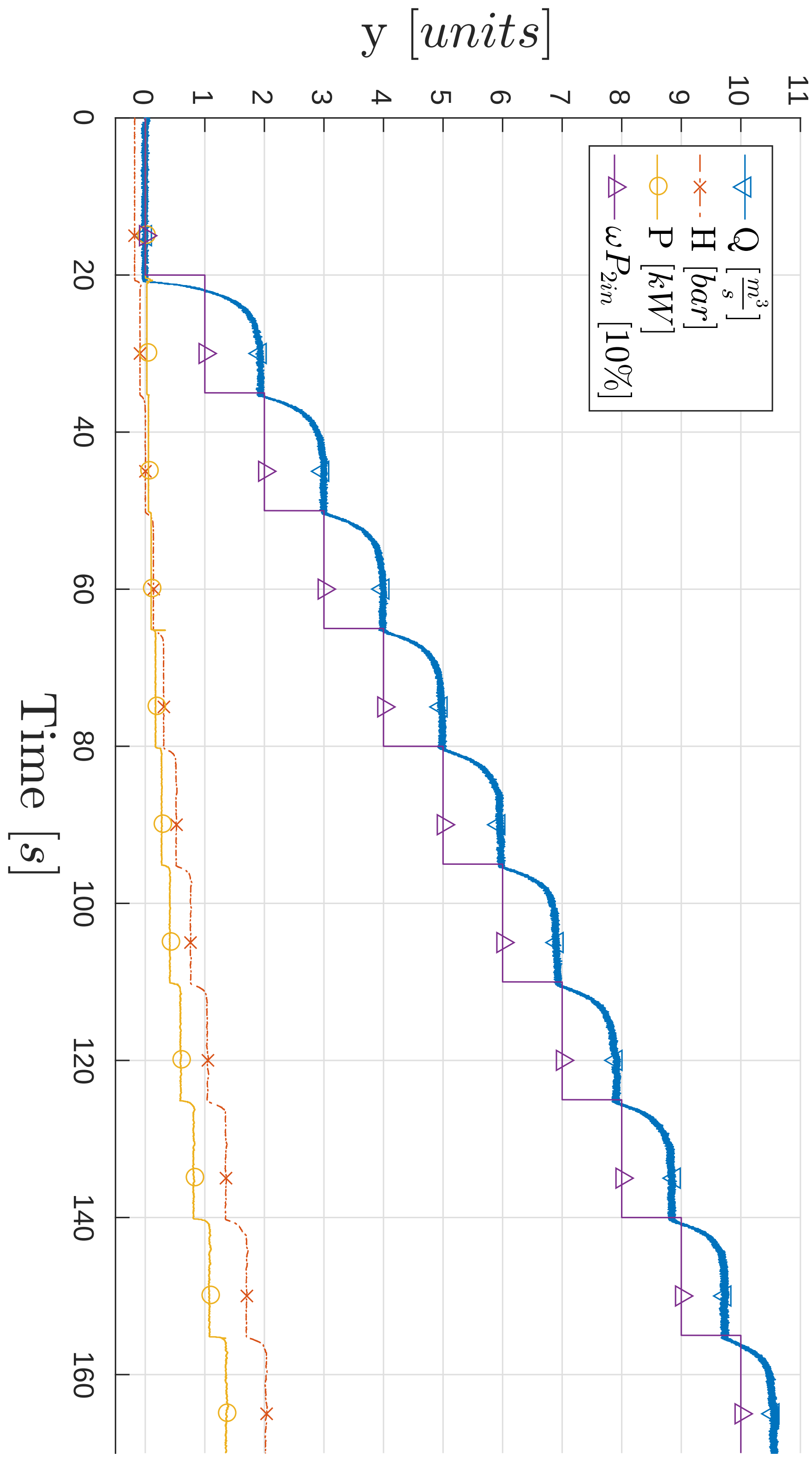

Figure D.4: Q, H, P and stair input ωP_{2in} at $CV_1 = 70\%$