



Norwegian University of  
Science and Technology

# Sensor-less Length Displacement Control of a Shape Memory Alloy Actuator

**Ulrik Utheim Sverdrup**

Master of Science in Cybernetics and Robotics

Submission date: June 2016

Supervisor: Amund Skavhaug, ITK

Co-supervisor: Jon Reidar Mathiassen, SINTEF Fiskeri og Havbruk AS

Norwegian University of Science and Technology  
Department of Engineering Cybernetics



**Title:** Sensor-less length displacement control of a Shape Memory Alloy Actuator

**Student:** Ulrik Utheim Sverdrup

### **Problem description:**

SINTEF Fisheries and Aquaculture AS (SINTEF) developed this problem as a smaller part of a large research project regarding humanoid robotics, where gripping with tactile sensors actuated by Shape Memory Alloy wires to mimic human muscle contractions as one area of focus.

This thesis aims to develop a system for self-sensing and a sensor-less length displacement control of a Shape Memory Alloy (SMA) actuator. By sensor-less it is meant no external sensors. The type of SMA actuator used throughout this thesis is from FLEXINOL<sup>®</sup> and is called MuscleWire<sup>™</sup>.

The main tasks of this project revolves around developing the following parts of the system:

- Designing an electrical circuit for measuring the varying resistance in the SMA actuator.
- Using an Arduino for generating a PWM signal to input in the SMA actuator and the measuring of the resistance.
- Use LabVIEW to program the system.
- Make an estimator which takes the resistance as input and outputs the displacement.
- Use a camera measurement for verification of the estimate.

Potential future tasks are:

- Include tactile sensors in feedback loop.
- Implement the solution in a gripper.

**Responsible professor:** Amund Skavhaug      NTNU

**Supervisor:** John Reidar Mathiassen      SINTEF





## Abstract

Motivated by designing a robotic gripper actuated by a Shape Memory Alloy actuator wire, this thesis investigates the possibility of designing a sensor-less length displacement controller by utilizing the inherent self-sensing ability of the wire. Where sensor-less means no external sensors. Based on work done by Liu et al. (2009), Dutta and Ghorbel (2005) and others in the field, the thesis further discusses the process of implementing the controller in a robotic hand or a gripper on an end effector of a robot arm.

Estimates based on step response measurement data were fitted with polynomial functions derived by least squares method both to compensate for poor measurement due to a fault in the circuit, and for estimating the displacement. The tests showed that the relation between the displacement and the electric resistance in the wire was close to linear in the contraction phase. Thus, a function was fitted to serve as an estimator of the displacement dependent on the calculated resistance. This estimate is then used as the feedback for the control loop.

Despite the poor performance of the estimator, the fast dynamics of the PID controller compensated for some of the violent and rapid perturbations, resulting in a tracking capability with relatively large error of the set point, but still tracking the set point to some extent. Further, the thesis discusses the applicability of the solution combined with a tactile sensor feedback despite the low accuracy of the estimator. Implementing a tactile sensor reduces the requirements with regards to accuracy because when the tactile sensor register a change in value it means the wire has contracted far enough. Thus, drawing the conclusion that the applicability is highly probable.

For future work with the system, the measurement must be improved. This thesis discusses the possible solution of designing a program which uses a *soft PWM* to ensure that the analog read is timed when the PWM signal is high. This removes the unwanted bias of the amplification and simplifies the circuit significantly.



## Sammendrag

*(Norwegian translation of the abstract)*

Motivert av å designe en robotgriper som aktueres av en Shape Memory Alloy aktuator wire, undersøker denne oppgaven muligheten for å designe sensorløs lengderegulator ved å utnytte den iboende egenskapen wiren har hvor den kan detektere sammentrekningen uten eksterne sensorer (på engelsk self-sensing"). Basert på arbeid utført av Liu et al. (2009), Dutta and Ghorbel (2005) og andre i feltet, diskuterer denne avhandlingen videre hvordan prosessen med å implementere regulatoren i en robotohånd eller en griper på en end-effector på en robotarm.

Estimator basert på måledata fra sprangresponsen til systemet ble tilpasset med polynomiske funksjoner med minste kvadraters metode, både for å kompensere for målefeil i kretsen, og for estimering av lengdeendring. Testene viste at forholdet mellom lengdeendringen og den elektriske motstanden i ledningen var nær lineær i sammentrekningsfasen. Dermed ble en funksjon tilpasset måledataen for å fungere som en estimator av lengdeendringen og videre anvendt som den tilbakekoblede prosessvariabelen i reguleringsløyfen.

Til tross for svake prestasjoner av estimatoren, klarte den raske dynamikken til PID-regulatoren å kompenserte for noen av de voldsomme og raske forstyrrelsene. Dette resulterte i en sporingsevne med forholdsvis stor feil i forhold til settpunktet, men som fortsatt klarer å følge det til en viss grad. Videre drøfter avhandlingen anvendelsen av løsningen kombinert med en taktil sensor som en ekstra sensor avlesning i tilbakekoblingen, til tross for den lave nøyaktigheten fra estimatoren. Implementeringen av en taktil sensor reduserer kravene til nøyaktighet på grunn av dersom den taktile sensoren registrerer en endring i verdi har wiren trukket seg nok sammen. Dermed trekkes den konklusjonen at anvendeligheten er svært stor.

For videre arbeid med systemet må målingene forbedres. Denne oppgaven diskuterer som mulig løsning å designe et program som benytter en *Soft PWM* for å sikre at *analoge read* funksjonen timet med når PWM-signalet er høyt. Dette fjerner uønsket bias i forsterkningen og forenkler kretsen betydelig.



## Preface

In the spring semester in 2016 in the fifth grade at NTNU, Trondheim, the students carries out their master project and thesis period. This is a project that is supposed to summarize the accumulated knowledge the students have acquired over their studies. The project results in 30 credit points and is therefore the only mandatory subject most students have this semester.

The author of this report, Ulrik Utheim Sverdrup, has throughout this project period also carried out all the practical tasks, such as designing the circuit and tests setup, as well as programming the LabVIEW code. The work has been done in collaboration with the responsible professor at the Faculty of Information Technology, Mathematics and Electrical Engineering at the Department of Engineering Cybernetics, Amund Skavhaug and external research scientists at SINTEF Fisheries and Aquaculture AS John Reidar Matthiassen.

This project is developed in cooperation with SINTEF. They have therefore contributed with needed equipment together with NTNU. In the big picture this project is a small part of a big ideology they are working towards. SINTEF currently are committed to a research project regarding humanoid and adaptive robotics, including griping with tactile sensors and kinetic motion actuated by SMA wires as one area of focus.

The aim of this thesis is to contribute with the author's accumulated knowledge and to educate the reader about the potential challenges regarding the implementation of a system for sensor-less control of a SMA actuator wire.

A big thanks goes out to the responsible professor at NTNU, Associate Professor at the Department of Engineering Cybernetics Amund Skavhaug for guidance, support and motivational meetings. Additionally, an equally big thanks goes out to the external supervisor John Reidar Matthiassen and researcher Elling Ruud Øye at SINTEF for excellent guidance, support and practical contributions throughout the project period.

*Ulrik Utheim Sverdrup*

---

*Ulrik Utheim Sverdrup  
Trondheim, june 2016*



# Contents

<b>Abstract</b>	<b>i</b>
<b>Sammendrag</b>	<b>iii</b>
<b>Preface</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>Symbols</b>	<b>xiii</b>
<b>List of Acronyms</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 History and motivation . . . . .	1
1.2 Equipment . . . . .	2
1.3 Overview of the chapters . . . . .	2
<b>2 Theory</b>	<b>3</b>
2.1 Literature review . . . . .	3
2.2 Background material . . . . .	5
2.2.1 FLEXINOL <sup>®</sup> Actuator Wire . . . . .	5
2.2.2 Arduino Uno . . . . .	7
2.2.3 LabVIEW . . . . .	7
2.2.4 Experimental setup - circuit . . . . .	8
2.2.5 PID controller . . . . .	11
2.2.6 Duhem Differential Model . . . . .	12
2.2.7 Feedforward control . . . . .	15
2.2.8 Reference model . . . . .	16
<b>3 Method</b>	<b>17</b>
3.1 The control-loop . . . . .	17
3.2 Experimental setup . . . . .	19
3.2.1 The rig . . . . .	20
3.2.2 The electrical circuit . . . . .	21
3.3 Arduino and LabVIEW code . . . . .	24
3.4 Mathematical model . . . . .	30

<b>4</b>	<b>Testing and results</b>	<b>31</b>
4.1	Open loop test . . . . .	32
4.1.1	Estimate of $V_{tot}$ . . . . .	32
4.1.2	Amplification compensator . . . . .	34
4.1.3	Estimate of the displacement . . . . .	37
4.1.4	Step response . . . . .	42
4.2	PID test . . . . .	50
4.2.1	Estimate feedback . . . . .	51
4.2.2	Actual $D_S$ feedback . . . . .	56
<b>5</b>	<b>Discussion</b>	<b>61</b>
5.1	PID test performance . . . . .	61
5.2	Spikes in estimate . . . . .	65
5.3	The mathematical model of hysteresis . . . . .	66
5.4	Measurement timing . . . . .	67
5.5	LabVIEW code . . . . .	68
5.5.1	Displacement estimate . . . . .	69
5.6	The circuit . . . . .	71
5.6.1	Low pass filtering and amplification error . . . . .	72
5.7	The total solution . . . . .	74
5.8	Application potential . . . . .	75
<b>6</b>	<b>Conclusion</b>	<b>77</b>
6.1	Future work . . . . .	78
	<b>References</b>	<b>79</b>
	<b>Appendices</b>	
<b>A</b>	<b>Flow diagram</b>	<b>83</b>
<b>B</b>	<b>LabVIEW - Front panel</b>	<b>85</b>
<b>C</b>	<b>LabVIEW - Block diagram</b>	<b>87</b>
<b>D</b>	<b>Camera image</b>	<b>89</b>
<b>E</b>	<b>Photos of the solution</b>	<b>91</b>



# List of Figures

2.1	$\xi$ denotes the degree of contraction, where 0 is fully elongated and 1 is fully contracted. T denotes the temperature. . . . .	5
2.2	An exaggerated illustration of how the size change with temperature change. . . . .	6
2.3	Schematic of the differential amplifier ( <i>left</i> ) and an operational amplifier ( <i>right</i> ). . . . .	10
2.4	Schematic of a first order passive low-pass RC filter. . . . .	11
2.5	Block diagram of the PID feedforward loop with $D_{S,est}$ as feedback. Designed in <i>draw.io</i> (JGraph, 2016). . . . .	15
3.1	Block diagram of the feedback loop. Designed in <i>draw.io</i> JGraph (2016). . . . .	17
3.2	Block diagram of the “ <i>Self-sensing and Estimator</i> ”-block from the feedback loop. Designed in <i>draw.io</i> JGraph (2016). . . . .	19
3.3	Schematic of the final experimental circuit. Designed in <i>DigiKey</i> Digikey (2016). . . . .	23
3.4	Image of the saved data (left) and the first saved image of the instant the LED was turned on (right), indicating the PWM signal was on. . . . .	26
3.5	$V_{R_W}$ , $\tilde{I}_C$ and $\tilde{R}_W$ plotted together to illustrate the cause of the spike in $R_W$ . Some of the values are scaled for easier comparison ( $\tilde{I}_C = 50 \cdot I_C$ and $\tilde{R}_W = R_W/4$ ). . . . .	28
3.6	$V_{R_W}$ , $\tilde{I}_C$ and $\tilde{R}_W$ plotted together to illustrate the cause of the spike in $R_W$ . Some of the values are scaled for easier comparison ( $\tilde{I}_C = 50 \cdot I_C$ and $\tilde{R}_W = R_W/4$ ). . . . .	29
4.1	Measured and calculated amplification with a polynomial fitted function. . . . .	36
4.2	The hysteresis characteristics of the wire. . . . .	38
4.3	A polynomial fitted function of the $D_{S,est}$ with respect to the resistance ( $R_W$ ). . . . .	39
4.4	Step response of the actual displacement ( $D_S$ ) compared to the estimated displacement ( $D_{S,est}$ ). . . . .	40
4.5	Bar graph of the error between $D_S$ and $D_{S,est}$ . . . . .	41
4.6	Step response of $V_{R_W}$ and $I_C$ . The values have been scaled to compare responsiveness. ( $\tilde{D}C = DC/4$ and $\tilde{I}_C = 50 \cdot I_C$ ). . . . .	43
4.7	Step response of $R_W$ . The values have been scaled to compare responsiveness ( $\tilde{D}C = 3 \cdot DC$ ). . . . .	44
4.8	Step response of $V_{R_W}$ and $I_C$ . The values have been scaled to compare responsiveness. ( $\tilde{D}C = DC/4$ and $\tilde{I}_C = 50 \cdot I_C$ ). . . . .	45

4.9	Step response of $R_W$ . The values have been scaled to compare responsiveness ( $\tilde{DC} = 3 \cdot DC$ ). . . . .	46
4.10	Step response of $D_S$ . The values of the duty cycle have been scaled to compare responsiveness ( $\tilde{DC} = DC/2$ ). . . . .	47
4.11	Plot of the actual range of the voltage drop ( $V_{R_k}$ ). . . . .	48
4.12	The PWM signal changing when the wire heats up. . . . .	49
4.13	PID-test1: $D_S$ tracking of set point with $D_{S,est}$ as feedback. . . . .	52
4.14	PID-test1: $D_{S,est}$ tracking of set point with $D_{S,est}$ as feedback. . . . .	53
4.15	PID-test2: $D_S$ tracking of set point with $D_{S,est}$ as feedback. . . . .	54
4.16	PID-test2: $D_{S,est}$ tracking of set point with $D_{S,est}$ as feedback. . . . .	55
4.17	PID-test3: $D_S$ tracking of set point with actual $D_S$ as feedback. . . . .	56
4.18	PID-test3: $D_{S,est}$ tracking of set point with actual $D_S$ as feedback. . . . .	57
4.19	PID-test4: $D_S$ tracking of set point with actual $D_S$ as feedback. . . . .	58
4.20	PID-test4: $D_{S,est}$ tracking of set point with actual $D_S$ as feedback. . . . .	59
5.1	Block diagram of the feedback loop with $D_S$ as feedback. Designed in <i>draw.io</i> (JGraph, 2016). . . . .	63
5.2	The step in duty cycle inputted to the estimate filtered. . . . .	65
5.3	Plot of the timing interval of measuring the analog read. Red rectangle marks measurement area. . . . .	67
5.4	The flow of the sending and receiving of the values. The orange blocks represent the LabVIEW code and the blue block represent the Arduino code. . . . .	69
5.5	PID-test3: $D_S$ tracking of set point with $D_{S,est}$ as feedback and the estimate is in this case filtered. . . . .	70
5.6	Equivalent schematic of the circuit when PWM signal is low. Designed in <i>DigiKey</i> Digikey (2016). . . . .	73
A.1	Color explanation for figure A.2. . . . .	83
A.2	Total flowdiagram of the LabVIEW code. . . . .	84
B.1	Screenshot from the “ <i>control section</i> ” in the front panel of the LabVIEW code. . . . .	85
B.2	Screenshot from the “ <i>observation section</i> ” in the front panel of the LabVIEW code. . . . .	86
C.1	The bolck diagram of the LabVIEW code. . . . .	87
C.2	The bolck diagram of the LabVIEW code. . . . .	88
D.1	Camera image with ROI (green square) and detected edge (red line). The axes are added after the image is captured for reference. . . . .	90

E.1	Image of the camera used to make the measurements of the actual displacement ( $D_S$ ). . . . .	91
E.2	The actual test rig (to the left) compared to the <i>CAD</i> -sketch (to the right). . . . .	92
E.3	Image of the wire with the tape for the edge detector and a 96 <i>gram</i> weight. View from the camera. . . . .	93

## List of Tables

4.1	The measurement of $V_{tot}$ is done with a multimeter and the calculation is done in the LabVIEW code. The estimate is the output of the fitted polynomial function. . . . .	33
4.2	The measurement of $V_{R_k}$ is done with a multimeter and the Raw data is directly from the Arduino, displayed in the LabVIEW code. The values in the parenthesis are based of measurements with multimeter because the value exceeded the upper value for the Arduino ADC. . . . .	35



# Symbols

## Symbols

$\beta_1$	Beta-value of the first transistor in the Darlington transistor.
$\beta_2$	Beta-value of the second transistor in the Darlington transistor.
$\beta_D$	Beta-value of the Darlington transistor.
$\xi$	Degree of contraction.
$A_3$	Analog input of Arduino.
$A_{diff}$	Amplification of the differential amplifier.
$A_f$	Austenite finish.
$A_{op}$	Amplification of the operational amplifier.
$A_S$	Austenite start.
$C_{LP}$	Capacitor of the Low-Pass filter.
$D_S$	Actual displacement measured with camera.
$D_{S,est}$	Estimated displacement.
$f_c$	Cut-off frequency of the Low-Pass filter.
$I_C$	Current through collector of Darlington transistor.
$M_f$	Martensite finish.
$M_S$	Martensite start.

$R_k$	Known resistor.
$R_{LP}$	Resistor of the Low-Pass filter.
$R_W$	Resistance of SMA actuator wire.
$SP$	Set Point.
$V_{BE}$	Base-Emitter voltage of the Darlington transistor.
$V_{CE}$	Collector-Emitter voltage of the Darlington transistor.
$V_{dd}$	Supply voltage.
$V_{in+}$	Non-inverting input of the differential amplifier.
$V_{in-}$	Inverting input of the differential amplifier.
$V_{R_k}$	Voltage drop over known resistor.
$V_{R_W}$	Voltage drop over SMA actuator wire.
$V_{tot}$	Voltage drop over known resistor and SMA actuator.

# List of Acronyms

## List of Acronyms

**ADC** Analog to Digital Converter.

**DAQ** Data Acquisition.

**GPIO** General-Purpose Input/Output.

**GUI** Graphical User Interphase.

**NI MAX** National Instruments - Measurements & Automation Explorer.

**NTNU** Norwegian University of Science and Technology.

**PID** Proportional-Integral-Derivative controller.

**ROI** Region Of Interest.

**SINTEF** SINTEF Fisheries and Aquaculture.

**SMA** Shape Memory Alloy.

**SMC** Sliding Mode Control.

**TDE** Time Delay Estimation.

**VISA** Virtual Instrument Software Architecture.





# Chapter 1

## Introduction

This thesis seeks to develop a system for sensor-less length displacement control of a SMA actuator. By sensor-less it is meant no external sensors. For potential future tasks the system will be implemented in a robotic gripper in an end effector to optimize space and weight on the robotic arm.

Shape memory alloys (SMAs) are a type of metallic alloy which possesses the ability to recover from length deformations of up to 5 - 6%. Through temperature change or exerted stress the crystal structure of the metal changes, allowing it to stretch and contract much like a muscle fiber.

The SMA actuator is known by many names. In this thesis the following names are used to describe it: SMA actuator, wire, actuator wire, SMA actuator wire and muscle wire. Additionally, the term *DC* is used to describe both *duty cycle* and *direct current*, but will be clarified before it is used. This is done to maintain a diverse language and improve the flow of the text for the reader.

### 1.1 History and motivation

Robots are becoming increasingly utilized in industry today and the allocated tasks for the robots are also becoming more sophisticated. Therefore, the requirements of the technology that controls the robots are increasing. A point of weakness in grippers in robotics today is their lack of feedback in the applied grip force. This can result in a gripper that is "*squeezing blind*" and risk losing or even destroying the object that is being gripped. Because of this, the gripper is less apt to pick up soft and uneven objects than for example a human. The mechanical actuators more commonly used for gripping such as linear actuators, servos and step motors, are also often large, therefore it can be awkward and difficult to install them on small objects.

The the main project at SINTEF is designing a gripper that will be actuated by Muscle Wires, which is a kind of wire that changes length proportionally to temperature. This can be enabled by setting an electric charge to it. The wire possesses many of the same properties as human muscle fibers, hence the name *Muscle Wire*<sup>TM</sup>. In addition, the gripper will be equipped with tactile sensors in the "*fungertips*" so that it can sense how hard it engages the object and adjust the gripping force accordingly.

The thesis is limited to the work with the SMA actuator. The tactile sensors are a part of the future work of this project which will be continued by SINTEF Fisheries and Aquaculture AS. However, the applicability of these sensors will be discussed.

## 1.2 Equipment

This is the equipment used in this project period.

- PicoScope 3000 Series, 2 channels (Oscilloscope)
- Gwinstek GPS 3030DD, Laboratory DC Power Supply
- FLEXINOL<sup>®</sup> Actuator Wire (150 $\mu$ m diameter)
- Arduino Uno
- Breadboard
- Multimeter
- Point Grey camera

As well as the electrical components such as: *LM392N* operational amplifier and a *TIP120* Darlington transistor.

## 1.3 Overview of the chapters

The remainder of this thesis is organized as follows. The *Theory* chapter (chapter 2) presents the articles on which this thesis is based, and it gives a quick introduction to the theory used throughout the project period. In the *Method* chapter (chapter 3) the different aspects of the solution to the project will be presented and the process of arriving at the final solution will be mentioned. Next, the *Testing and results* chapter (chapter 4) will explain and present the tests and their corresponding results. After that, the different subjects mentioned throughout the thesis are discussed in chapter 5, followed by the author's conclusions and a presentation of the remaining future work in chapter 6. Lastly, after the *References*, the *Appendix* shows different images, code snips of the LabVIEW front panel and block diagram, respectively, and the photos of the system solution.

Some subjects will be discussed briefly along the way to enlighten the reader of a subject which will be discussed in detail in the discussion chapter.

The reader is expected to have some knowledge about basic control theory. Although the subjects of chapter 2.2 may seem banal to some, it is meant as a supplement for the theory used in throughout the project period.

This thesis is preferably printed in colors to distinguish between plots and such that the plots and images are easier to interpret.

# Chapter 2

## Theory

Firstly, in this chapter is a “*Literature review*”. This consists of a summary of a few articles based on similar work and research. This is done to give the reader a quick overview of the “*state of the art*”. Also, the relevance of the article will be shortly discussed.

Next, comes a section of descriptions of theory used throughout this thesis, including equations and explanation of the principle behind the theory.

### 2.1 Literature review

In the following paragraphs, the literature that forms the conceptual basis of the work done throughout this thesis are summarized and discussed.

In the article by Liu et al. (2009) mathematical models are deduced for an *inverse compensator* used to compensate for the dynamics of the SMA actuator by feedforward. Within this *inverse compensator* is a inverse hysteresis model to compensate for the highly nonlinear hysteresis characteristics of the SMA actuator. Additionally, the temperature dynamics of the SMA actuator are calculated, and a spring deformation model used to calculate the electrical power needed to change the temperature of the wire as well as the work required to provide elastic energy to the bias spring. The *inverse compensator* outputs the required duty cycle for the PWM signal to reach the desired displacement. Lastly a model for the self-sensing property of the SMA actuator is used as feedback to close the main control loop.

The article by Liu et al. (2009) was meant to form the backbone of this thesis, but because of the complexity and magnitude of the mathematical model, it proved to be too advanced for a single person to reach the same level of understanding of the mathematical models derived at this amount of time. However, the article served as a good example and inspiration regarding the design of the experimental electrical circuit and the principle of the control loop.

In the article by Lee et al. (2013) a simple, robust and model-free tracking control of the SMA actuator where the core principle is to estimate unknown dynamics and disturbances. This was done by using the input-output relationship of the closed-loop system with intentionally time-delayed information to compensate for the hysteresis

and the other nonlinearities in the SMA actuator dynamics. This estimator is called a Time Delay Estimation (TDE). They also define a nonlinear sliding surface with hyperbolic tangent function, called Sliding Mode Control (SMC), in an attempt to alleviate the negative effects resulting from velocity saturation of the SMA actuator and noise due to numerical differentiation.

Although Lee et al. claims to implement a simple tracking controller it appears quite complex for one person to overcome alone in a short period of time. Additionally, the experimental setup in this article is quite different with respect to design compared to the planned experimental setup for this thesis. Nevertheless, this is a proof that a mathematical model expressing the physics of the different dynamics of the SMA actuator is not necessary to design a precise tracking controller.

The third article written by Senthilkumar et al. (2011) develops a system which is based on a PWM adaptive position controller for the SMA actuator. Initially an experimental open loop test is conducted to determine the transfer function of the SMA actuator. The experimental setup has a few external sensors measuring factors such as temperature, displacement, load and air velocity. Additionally the ambient temperature was considered negligible as the SMA actuator contracts only after reaching austenite start  $A_S$  temperature (explained in chapter 2.2.1). Through testing the tracking abilities of a few different types of adaptive controllers they found that in spite of disturbances in temperature, the adaptive and modulated adaptive controllers were able to track the change in set point. Additionally, they found that improved accuracy is obtained with modulated adaptive controllers.

The transfer function derived in Senthilkumar et al. (2011) could have been used in this thesis if it were discovered in the earlier stage of the project. In hindsight, it would have been preferable to implement a model for the SMA actuator, regardless of the complexity, to compensate in some matter for the characteristics of the actuator. However, an essential focus of this thesis was to have a sensor-less solution and utilize the self-sensing capabilities of the wire only. Therefore, the experimental tests are not entirely applicable to this thesis.

Lastly, in an article by Ma and Song (2003), a comparative study of three different controllers based on PD (Proportional Derivative) controller and a PWM module was conducted. The goal of the article was to design a controller which reduce the energy consumption by the SMA actuator by using a PWM signal.

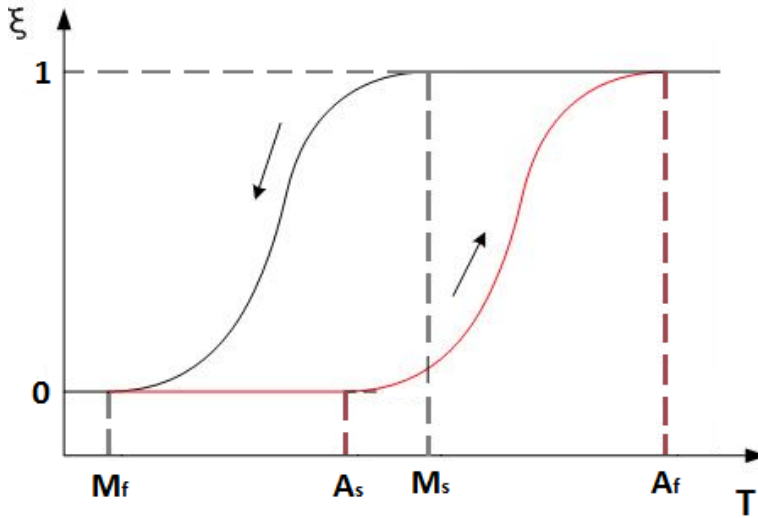
Seeing as the *Digilent LINX - LabVIEW Makerhub* package (see chapter 2.2.3) has a limited functionality, this solution would be difficult to implement because it would require the system to be able to change the base frequency of the PWM signal, which was restricted in this package.

## 2.2 Background material

In this chapter the different theory used in the different aspects in this thesis is briefly summarized.

### 2.2.1 FLEXINOL<sup>®</sup> Actuator Wire

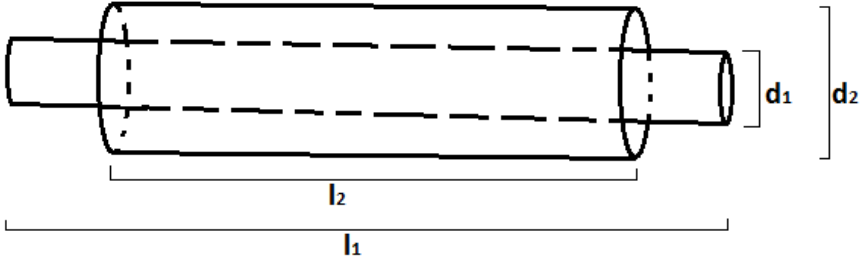
The SMA actuator is a metal wire made up of a *Nickel-Titanium (NiTi) alloy* which gives it the ability to contract like a muscle fiber when heated. It performs a reversible solid state phase transformation, called a *martensitic transformation*. The name comes from the two phases it changes between when the length displacement happens, which are *martensite* and *austenite*. *NiTi* alloys change from martensite to austenite when heated where  $M_S$  and  $M_f$  denotes the temperature at which the transition to martensite starts and finishes, consequently causing the elongation. Similarly,  $A_S$  and  $A_f$  denotes the temperatures at which the phase transforms to austenite, resulting in the contraction of the wire. This is illustrated in figure 2.1.



**Figure 2.1:**  $\xi$  denotes the degree of contraction, where 0 is fully elongated and 1 is fully contracted.  $T$  denotes the temperature.

Repeated use can cause a shift in the transformation temperatures. This is due to a phenomenon called *functional fatigue*. In principle this is the same as *metal fatigue*, but describes in this case, the cumulative damage the crystal structure of the metal takes with cyclic use, ie. repeated contractions. The SMA actuator from Flexinol<sup>®</sup> is said to withstand "... *tens of millions of cycles...*", according to Dynalloy (2016) if the recommended approximated limitations are upheld.

When the temperature increases the electrical resistance decreases proportionally to the displacement of the length. This relation is used to estimate the displacement of the wire without using external sensors. The SMA actuator is therefore said to have a “*self-sensing*” ability (Liu et al., 2009). When the wire contracts the three-dimensional volume of the wire is approximately maintained, as illustrated in figure 2.2 and explained in equation 2.1. Thus, as the length shortens the diameter increases. These two are the dominant factors for the drop in resistance when the wire contracts.



**Figure 2.2:** An exaggerated illustration of how the size change with temperature change.

$$\begin{aligned} \pi \left(\frac{d_1}{2}\right)^2 \cdot l_1 &\approx \pi \left(\frac{d_2}{2}\right)^2 \cdot l_2 \\ d_1^2 l_1 &\approx d_2^2 l_2 \end{aligned} \quad (2.1)$$

Where  $l_2$  and  $l_1$  is the length before and after the contraction, and  $d_1$  and  $d_2$  is the diameter before and after contraction, respectively.

For the SMA actuator to maintain its displacement range an opposing force is used to stretch the wire back to its original length while cooling. If there is no force exerted in the cooling phase, the wire will not elongate much. Correspondingly, the actuator will not contract much either. This opposing force is called a *bias force*. With a “*dead weight*” bias force, which is a weight hanging at the end of the wire, displacement can vary from 4 - 6% (Dynalloy, 2016). The range of the displacement increases if the wire is woven into a rotating helix explained in an article by Grant and Hayward (1997). However, this exceeds the limitations of this thesis.

The SMA actuator is heated in accordance with the *Lenz-Joule law* described in equation 2.2 below (Senthilkumar et al., 2011).

$$W = I_C^2 \cdot R_W = I_C \cdot V_{R_W} = \frac{V_{R_W}^2}{R_W} \quad (2.2)$$

Where in this case  $I_C$  is the current of the collector in the Darlington transistor, and also the current through the SMA actuator.  $R_W$  is the resistance in the actuator (' $W$ ' is for "wire") and  $V_{R_W}$  is the voltage drop over the actuator.

### 2.2.2 Arduino Uno

Arduino Uno is a microcontroller board based on the 8-bit *ATmega328P* with a 16 MHz crystal core. It has 14 digital General-Purpose Input/Output (GPIO) ports where 6 of them can be used as PWM output and 6 as analog inputs. Additionally, it has the following communication protocols: *SPI*, *USART*, *I2C*. It has a 10-bits Analog to Digital Converter (ADC) giving it 1024 steps and a sensitivity of 3.4 mV (Arduino, 2016).

Although the system in this project only uses one PWM output pin and one analog input pin, this microcontroller was chosen because of its compatibility with the Digilent LINX - LabVIEW Makerhub package. This package is described in chapter 2.2.3.

### 2.2.3 LabVIEW

LabVIEW is the program used for programming this system. The reason being that it is the preferred programming language at SINTEF. The communication with the Arduino was done by installing a package called *Digilent LINX - Makerhub for LabVIEW*. This is mentioned below.

The standard in LabVIEW called Virtual Instrument Software Architecture (VISA) includes the software libraries for applications such as serial communication, which is the one used for earlier versions of the LabVIEW solution in this thesis.

A different driver software package used is called National Instruments Image Acquisition Software (NI IMAQ) and is used to acquire images from a *Point Grey* USB camera.

#### Least squares polynomial fit

In the LabVIEW program a block called *General Polynomial Fit VI* is used to make a function to estimate values such as contraction, amplification and voltage drop. As default this is set to use the *Polynomial Least Squares method* for fitting a function. In this case the method fits a polynomial function of a chosen order that approximates a complicated function. The function block in LabVIEW outputs the best fitted line and the coefficients of the polynomial function.

The method works by finding the sum of the squared vertical deviation from the data points. Then proceeds by minimizing this sum (Weisstein, 2016).

### Digilent LINX - LabVIEW Makerhub

LINX is an open source project by *Digilent* which is designed to simplify the development of embedded applications on a platform such as Arduino using LabVIEW. It compiles the LabVIEW code to an executable code and builds it on the corresponding Arduino. It is compatible with both the Uno and the Mega2560 within the Arduino family.

It uses a USB serial communication protocol between the Arduino and the LabVIEW program. One difference in launching the code with the LINX add-on compared to just a regular LabVIEW program is a delay of a few seconds before the program runs. This is due to the code being uploaded and built on the Arduino. However, when the code runs the delay between sending commands from the LabVIEW front panel and the Arduino actually executing the command is neglected. This is discussed in chapter 5.5.

The LINX VIs in the LabVIEW code is sending commands to the LINX firmware running on the Arduino connected by the USB port (National Instruments, 2016b). On LabVIEW MakerHub's own homepage (National Instruments, 2016a) they claim that when using the serial interface the execution speed at which the LINX VIs operate is  $\sim 4$  milliseconds per LINX VI / Packet. Where the *Packet* is the PWM duty cycle for the *Set Duty Cycle* VI and the analog read value for the *Analog Read* VI. In this code there is only one package for each of the two LINX VIs.

### Edge detection

The *IMAQ Find Edge VI* in LabVIEW is used for finding straight edges (lines) within a region in an image inputted into the block. The *Options* input of the block specifies the search direction within the Region Of Interest (ROI), in this case from left to right. It outputs the coordinates for the start and the stop point of the detected line denoted  $(X_1, Y_1)$  and  $(X_2, Y_2)$ , respectively. This is illustrated in figure D.1.

#### 2.2.4 Experimental setup - circuit

In the following paragraphs are some short descriptions of the most significant components used in the experimental setup. It is not essential for the reader to read this, but it is advantageous to have knowledge about the components to some degree.

### Darlington transistor

A Darlington transistor of the type *TIP120* is comprised of a pair of bipolar transistors in a configuration where the current amplified by the first transistor is further amplified by the second one. This results in a much higher current gain than what each transistor could have supplied separately, however it behaves like a single



transistor. The gain is given by the *beta factor* ( $\beta$ ) of each individual transistor in the Darlington configuration ( $\beta_1$  and  $\beta_2$ ).

$$\beta_D = \beta_1 \cdot \beta_2 + \beta_1 + \beta_2 \quad (2.3)$$

If the *beta factor* of the two transistors that make up the Darlington driver are large enough (in the hundreds), the equation can be approximated with the following:

$$\beta_D \approx \beta_1 \cdot \beta_2 \quad (2.4)$$

Although the Darlington transistor has its advantages it also have some drawbacks. One of them is the *Base-Emitter* voltage ( $V_{BE}$ ). Since a Darlington transistor comprises two transistors the *Base-Emitter* voltage will be the sum of  $V_{BE}$  from the two transistors ( $V_{BE1}$  and  $V_{BE2}$ ), which is about two diode drops. However, this has no effect on the experimental setup for this thesis. Since the *TIP120* used is a *NPN-transistor* the voltage  $V_{BE}$  goes to ground when the PWM signal goes low. When the Darlington transistor is completely open/on the *Collector-Emitter* voltage ( $V_{CE}$ ) is about 0.75 V.

### Differential and operational amplifier

A differential amplifier (diff.amp.) is an analog amplifier which amplifies the difference between two input voltages following the formula for the output voltage of equation 2.5 when the resistors are arranged as illustrated in figure 2.3.

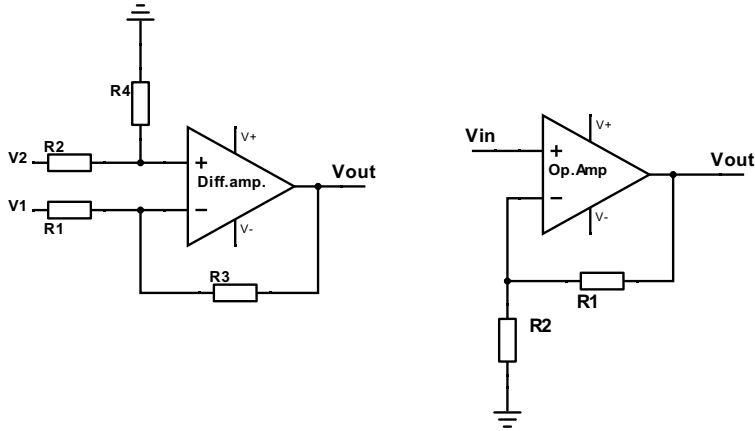
$$V_{out} = V_2 \left( \frac{R_4}{R_2 + R_4} \right) \left( \frac{R_1 + R_3}{R_1} \right) - V_1 \frac{R_3}{R_1} \quad (2.5)$$

Where  $V_1$  and  $V_2$  are the two input voltages and the resistors are arranged as depicted in figure 2.3. Furthermore, if  $R_1 = R_2$  and  $R_3 = R_4$  the expression can be simplified as shown in equation 2.6:

$$\begin{aligned} V_{out} &= \frac{R_3}{R_1} (V_2 - V_1) \\ &= A_{diff} \cdot (V_2 - V_1) \end{aligned} \quad (2.6)$$

Where the gain:

$$A_{diff} = \frac{R_3}{R_1} \quad (2.7)$$



**Figure 2.3:** Schematic of the differential amplifier (*left*) and an operational amplifier (*right*).

The type used in this thesis is of the type *LM392N* and is a single-supply operational amplifier (op.amp.). This means that it is sufficient to supply only a positive voltage to drive the op.amp. Another way of implementing an op.amp. is by amplifying with respect to ground. The output voltage ( $V_{out}$ ) is then calculated as such:

$$\begin{aligned} V_{out} &= \frac{R_1}{R_2}(1 + V_{in}) \\ &= A_{op} \cdot (1 + V_{in}) \end{aligned} \quad (2.8)$$

Where the gain is:

$$A_{op} = \frac{R_1}{R_2} \quad (2.9)$$

If the differential amplifier is arranged as a *Unity Gain Differential Amplifier* it means that all the resistors are of the same value, thus  $A_{diff} = 1$ .

### Low-Pass filter

In electronics a low-pass RC filter is created by a resistor, a capacitor and a load. The resistor is in series with the load while the capacitor is in parallel. The configuration is depicted in figure 2.4. This constitutes a first order RC-filter. The cutoff frequency

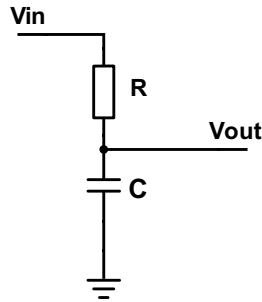
( $f_c$ ) is determined by the time constant of the RC term and is calculated by inserting the respective values in equation 2.10.

$$f_c = \frac{1}{2\pi \cdot R \cdot C} \quad (2.10)$$

Where  $R$  is the resistance value of the resistor in the filter ( $R_{LP}$  in this thesis) and  $C$  is the capacitance value of the capacitor in the filter ( $C_{LP}$  in this thesis). When a filter is implemented the signal is delayed for a small period of time depending on the  $RC$  term. This is due to the charging of the capacitor. The phase shift is described in equation 2.11.

$$\phi = -\arctan(2\pi fRC) \quad (2.11)$$

What signifies a Butterworth filter is that the pass band of the Bode plot is flat with a gain of  $0dB$ , and at the cut-off frequency ( $f_c$ ) the gain is  $-3dB$ . For a first order low-pass filter the slope of the gain is  $-20dB/decade$ .



**Figure 2.4:** Schematic of a first order passive low-pass RC filter.

### 2.2.5 PID controller

A Proportional-Integral-Derivative controller (PID) controller is a control loop feedback mechanism based on a continuously calculated error value. The error value is the difference between a chosen set point ( $SP$ ) and a measured or calculated process variable ( $pv(t)$ ). In the case of this thesis the process variable is the estimated displacement of the SMA actuator wire. The controller calculates a weighted sum

designed to minimize the error over time, thus bringing the measured or calculated process variable to the wanted set point. The control variable is calculated by the following equation (2.12):

$$U(s) = K_c \left( 1 + \frac{1}{T_i s} + T_d s \right) E(s) \quad (2.12)$$

Which means that  $K_c = K_p$ ,  $K_i = \frac{K_c}{T_i}$  and  $K_d = K_c T_i$ , and where  $e(t)$  is the error (equation 2.13) and  $K_p$ ,  $K_i$  and  $K_d$  denote the non-negative coefficients for the proportional ( $P$ ), integral ( $I$ ) and derivative ( $D$ ) terms respectively.

The coefficients weight the present, past and future values of the measured error, respectively. The  $P$  part multiplies the error with a constant ( $K_p$ ), thus if the error is large the  $P$  term will be large. The  $I$  part integrates the values over time, meaning that if the current output is not sufficiently large, the accumulated error will cause the  $I$  term to increase. Lastly, the  $D$  part differentiates the error with respect to time. This means that if the change of process variable and consequently the error is large, the  $D$  term will typically counteract this for future values, depending on the sign of  $K_d$ . The error is expressed as follows:

$$e(t) = SP - pv(t) \quad (2.13)$$

Where  $SP$  denotes the set point and  $pv(t)$  is the process variable which in this thesis is the estimated displacement ( $D_{S,est}$ ).

The model is designed such that it only needs the process variable and the set point as input, without any model of the plant. By tuning the three parameters  $K_p$ ,  $K_i$  and  $K_d$  the controller can bring the system to its desired set point. The response of the controller is characterized by factors such as overshoot, settling time, responsiveness to an error and the degree of oscillation.

Tuning of the parameters is important to get the desired performance. There are several techniques including manual tuning done by eye which requires some experience, and other commonly used methods for tuning the parameters are *Ziegler-Nichols*, *Skogestad's*, pole placement and different tuning software.

### 2.2.6 Duhem Differential Model

The Duhem model is an analytical approach to describe a smooth hysteretic behavior. It is widely used in mechanical and electrical engineering. In practice it is mostly used in "*black-box*" approaches, meaning by knowing the experimental input and

output data, the model finds the parameters such that the real output matches the experimental output data.

The model follows the fact that if an input,  $u(t)$ , is an increasing function, then the output  $v(t)$  increases along a one path. However, if  $u(t)$  is a decreasing function,  $v(t)$  decreases along a different path. The slope functions are often called  $g_-$  and  $g_+$  for the decreasing and increasing function, respectively.

The Duhem model is given by the equation:

$$\begin{cases} \dot{v} = g_+(u(t), v(t)) \cdot (\dot{u})^+ - g_-(u(t), v(t)) \cdot (\dot{u})^- \\ v(0) = v_0 \end{cases} \quad (2.14)$$

Where

$$(\dot{u})^\pm = \frac{|\dot{u}| \pm \dot{u}}{2} \quad (2.15)$$

And the slope functions:

$$g_{+/-}(u) = \frac{1}{\sigma_{+/-}\sqrt{2\pi}} \exp\left(-\frac{(u - \mu_{+/-})^2}{2\sigma_{+/-}^2}\right) \quad (2.16)$$

where  $\mu$  is the mean and  $\sigma^2$  is the variance, and the subscripts  $+$  and  $-$  denote the increasing and decreasing curves, respectively.

Then the output  $v$  representing the major loop is given by:

$$\begin{aligned} v_{+/-} = h_{+/-}(u) &= \int_{-\text{inf}}^u g_{+/-}(u') du' \\ &= \frac{1}{2} \left[ 1 + \text{erf}\left(\frac{u - \mu_{+/-}}{\sigma_{+/-}\sqrt{2}}\right) \right] \end{aligned} \quad (2.17)$$

Where *erf* is the *error function*.

However, since for instance the actuator wire does not necessarily contract or elongate from maximum to minimum each time the hysteresis must be expressed for both the long and the short displacement changes. These functions are called the major and minor loops, respectively.

The major loop is given by the differential equation:

$$\frac{dv}{du} = \begin{cases} \frac{1}{\sigma_+ \sqrt{2\pi}} \exp\left(-\frac{(u-\mu_+)^2}{2\sigma_+^2}\right), & \dot{u} \geq 0 \\ \frac{1}{\sigma_- \sqrt{2\pi}} \exp\left(-\frac{(u-\mu_-)^2}{2\sigma_-^2}\right), & \dot{u} < 0 \end{cases} \quad (2.18)$$

The minor loop functions can be expressed as the major loop scaled by a suitable constant  $n_{i_{+/-}} \in [0, 1]$  where  $i = 1, 2, \dots, N$ . This gives the minor loop slope function expressed by:

$$g_{i_{+/-}} = \frac{n_{i_{+/-}}}{\sigma_+ \sqrt{2\pi}} \exp\left(-\frac{(u-\mu_+)^2}{2\sigma_+^2}\right) \quad (2.19)$$

The scaling constant must be expressed by the couple  $(u(t), v(t))$  and the corresponding slope and major loop functions, and is given by:

$$\begin{aligned} n_{i_+}(u, v) &= \frac{h_-(u) - v}{h_-(u) - h_+(u)}, \quad \dot{u} \geq 0 \\ n_{i_-}(u, v) &= \frac{v + h_+(u)}{h_-(u) - h_+(u)}, \quad \dot{u} < 0 \end{aligned} \quad (2.20)$$

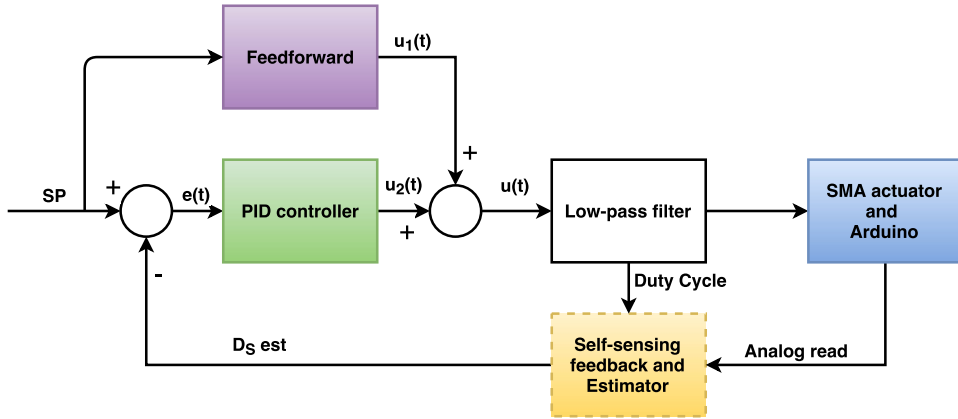
Resulting in the complete differential hysteresis model expressed as:

$$\frac{dv}{du} = \begin{cases} n_{i_+} \cdot g_+(u), & \dot{u} \geq 0 \\ n_{i_-} \cdot g_-(u), & \dot{u} < 0 \end{cases} \quad (2.21)$$

The theory behind the Duhem model for hysteresis is found in Visintin (2006), Visintin (2013), Dutta and Ghorbel (2005) and Naser and Ikhrouane (2013). It is also mentioned shortly in (Lee et al., 2013) and (Liu et al., 2009). All the articles are quite similar, but may use different signs for the different parameters. This summary is mainly based on Dutta and Ghorbel (2005).

### 2.2.7 Feedforward control

The feedforward controller is a method for rapid tracking of time varying trajectories or set points. This is done by inputting the set point into a feedforward block. Generally an inverse of the plant (in this case the SMA actuator) is used to compensate for the dynamics of the plant. Ideally, this gives an output which combined with the output of the PID controller comprises an input for the system with no deviation and perturbations. The flow of such a system if it were implemented in this thesis is illustrated in figure 2.5.



**Figure 2.5:** Block diagram of the PID feedforward loop with  $D_{S,est}$  as feedback. Designed in *draw.io* (JGraph, 2016).

This is essentially the way Liu et al. (2009) designed the controller, where the *feedforward* was called an *Inverse compensator*.

If the plant was some function  $G_p(s)$ , than the feedforward transfer function would be:

$$FF(s) = \frac{1}{G_p(s)} \quad (2.22)$$

The process variable is in this thesis the displacement estimate ( $D_{S,est}$ ).

### 2.2.8 Reference model

A reference model is a technique used for trajectory or set point generation. It is used to construct a feasible trajectory between the two set points such that the inputted set point changes more smoothly. The simplest form of a reference model is obtained by using a low-pass filter according to Fossen (2011).

$$\frac{D_{S,est}}{SP}(s) = h_{lp}(s) \quad (2.23)$$

where  $D_{S,est}$  is the desired state and  $SP$  denotes the reference signal or the set point.



# Chapter 3

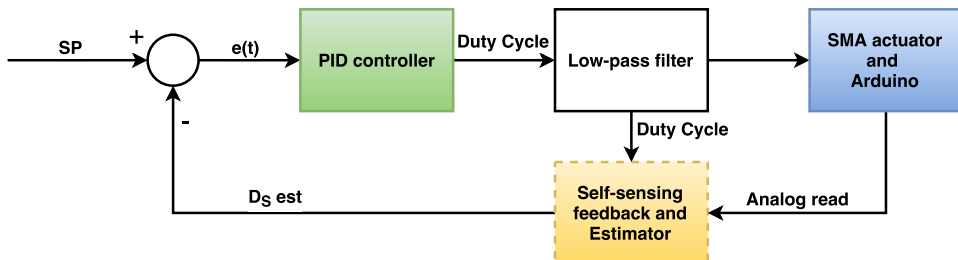
## Method

Following in this chapter is the process of arriving at the different aspects of the solution, such as describing the flow of the control loop, the experimental setup consisting of a physical rig and an electrical circuit. Then the development of the Arduino and LabVIEW code are described, and lastly, the attempt of implementing the mathematical model is described.

### 3.1 The control-loop

The final solution of this system is a simplified version inspired by several of the articles, including Liu et al. (2009), Lee et al. (2013), Dutta and Ghorbel (2005) among others. The logical flow diagram for the LabVIEW program is illustrated in figure A.2 with color coded blocks.

The system is based on a PID controller calculating the gain input for the SMA actuator. The output of this plant, is the *analog read* value from the Arduino which is the voltage drop over a known resistor ( $R_k$ ) in the circuit. This value is used as an input along with the filtered duty cycle to estimate the contraction ( $D_{S,est}$ ) which is then used as a feedback to generate the error input to the PID controller. This means that the only measured value in this system is the *raw data* of the *analog read*. The rest of the values used throughout the system is based on calculations from the *analog read* and is therefore considered to be estimates. The flow of the control loop is illustrated in figure 3.1.



**Figure 3.1:** Block diagram of the feedback loop. Designed in *draw.io* JGraph (2016).

The PID controller takes in the error ( $e(t)$ ) between the set point ( $SP$ ) and the estimated length displacement of the SMA actuator ( $D_{S,est}$ ), and calculates the input duty cycle ( $DC$ ) for the SMA actuator. However, due to spikes in resistance value ( $R_w$ ) as a result of a slow increase of the current ( $I_C$ ) compared to the quick response of the voltage ( $V_{R_w}$ ) as illustrated in figure 3.5 on page 28, the duty cycle gets filtered through a second order Butterworth low-pass filter. This attenuated the spikes enough to produce usable values for the estimator, but still keep the rapid response of the change in values for the duty cycle. To filter the duty cycle is considered a temporary solution and will be discussed in chapter 5.2 with regards to performance and consequences.

Nevertheless, the filtered duty cycle is then sent to the Arduino, which uses it to produce a PWM signal with the mentioned duty cycle to heat up the SMA actuator (in accordance to equation 2.2). This circuit reaches a maximum current of  $280mA$  and a voltage of  $14V$  when the wire is fully contracted, which is about  $45mm$ . Depending on the input voltage to the SMA actuator as a result of the duty cycle of the PWM signal, the wire will contract or elongate. In this system the contraction starts very slowly at about 60% duty cycle, but it is dependent on the ambient room temperature as well. This change in displacement changes the electric resistance through the wire and thus, the voltage drop over it.

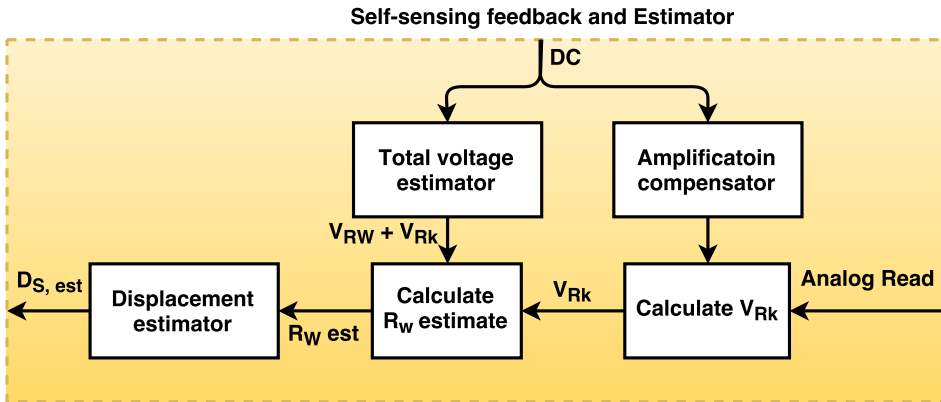
The low-pass filtered duty cycle is also sent to the *Self-sensing feedback and Estimator* block, but because of the low-pass filter implemented in the circuit the analog read from the Arduino is an average of the PWM signal considered as a DC (Direct Current) signal. Hence, a few measurements behaves differently than if the signal is treated as a momentary direct current where the Darlington transistor is fully open when the PWM duty cycle is high. The total voltage, which is the voltage drop over the known resistor and the wire ( $V_{tot} = V_{R_w} + V_{R_k}$ ), is originally considered to be constant at approximately  $19.25V$  when the signal is not filtered. However, by implementing the low-pass filter  $V_{tot}$  is now dependent on the duty cycle of the PWM signal.

Since the issue following the implementing of the low-pass filter was detected at a later stage of the project period and all the operational amplifiers were taken, an estimator were implemented to calculate the total voltage drop dependent on the duty cycle. The estimator is based on a step response test conducted prior to running the program, to make a dataset for fitting a polynomial function. This test is described in chapter 4.1.1.

Also, because of the filtering the amplification gets biased by the supply voltage when the PWM signal get low. This means that the amplification is biased more as the duty cycle gets lower. Therefore, similar to the estimate of the total voltage, the

estimated amplification value used for the compensator is based on measurements made prior to running the program. This is explained in chapter 4.1.2.

Lastly,  $R_W$  is used to fit a polynomial function with a measured contraction from an open loop test. Thus, making an estimator of the displacement which is dependent on the resistance in the wire. How this test was conducted is described in chapter 4.1. This results in an estimated displacement ( $D_{S,est}$ ) output which is used to compare with the set point and calculate the error input for the PID controller.



**Figure 3.2:** Block diagram of the “*Self-sensing and Estimator*”-block from the feedback loop. Designed in *draw.io* JGraph (2016).

To test the properties of the wire an experimental setup must be design and constructed. In the next chapter follows a description and a brief discussion about a variety of different suggestions to an experimental setup, both with regards to the rig and the electrical circuit.

## 3.2 Experimental setup

The term *experimental setup* covers the total physical package of the solution, including *the rig* consisting of a metal frame, a camera and the wire it self. Additionally, it covers the electric circuit with its components and the power supply.

In this section a few different experimental setups, that were investigated, will be explained and discussed. After that, the chosen experimental setup will be described. A setup like this should be able to measure the desired effects and characteristics with a high reliability and repeatability such that if needed the results can easily be reproduced.

### 3.2.1 The rig

By “*The rig*” it is meant the actual physical framework of the experimental setup, excluding the electrical circuit. The purpose of the rig is to hold the wire and camera in place, thus ensuring a reliable test setup.

#### Earlier suggested versions

The first suggested setup was based on the principle of expansion. If for instance the project would reach a phase where a protagonist/antagonist-based *bias-force* should be implemented it would be included in the setup without to much work. A setup like this means that a second wire is used to stretch the first wire which is doing the actual work like muscles work. Additionally, if a tactile sensors was to be tested, it could also easily be included in the design. However, since the scale of the project proved greater than expected this design was considered too complex compared to the benefits of implementing it and was eventually scrapped.

The second design was based on a setup done in an earlier project. The difference from the last one was the measurement of the displacement. This would be done with a short range IR proximity sensor. Where short range means approximately 4 - 30cm. However, considering the displacement percentage of 4 - 6% which results in 4 - 6cm with a wire of 1m, the accuracy of a sensor at a realistic price range is not sufficient. The requirements determined by the author of this thesis claims that worse accuracy than 0.5mm is not acceptable. Hence, this suggestion was also eventually scrapped.

The third rig was designed quite similar to the second one. The difference is the measurement of the displacement. The setup uses an optical encoder or a potentiometer to measure the change in angle with a negligible friction in the rotation. To cancel the slip caused by the wire displacement along the surface of the encoder, a nylon thread is wrapped around the rotary element of the encoder or potentiometer. This is because a nylon thread has very little elasticity. To calculate the displacement the angle of the encoder or potentiometer must be found. For the encoder it would be  $\theta \cdot r = h$ , where  $\theta$ ,  $r$  and  $h$  are the angle of the encoder, the radius of the rotary element on the encoder and the actual displacement of the wire, respectively. For the potentiometer the different resistance values would have to be mapped to a displacement length. This solution was scrapped because it was suggested late in the design period and the encoders available was not suitable for the task.

### Final solution

Lastly, the chosen design was the same as the one used in a project by Sverdrup in the fall of 2015. This rig is constructed with aluminum bars high enough that the wire can hang fully elongated with a weight at the end. Opposite to the wire is a bar for the camera which is pointing towards the wire in front of a ruler. This time however, the camera feed was used to measure the displacement, rather than just take pictures. The design is based on that the wire hangs at the top of a rig with a weight of 96 *grams* at the end, serving its purpose as the bias force. To serve as a reference point for the camera and the edge detector in the LabVIEW code, a piece of tape was attached at the bottom of the wire. The edge detector is depicted in figure D.1, and the rig is depicted in appendix E along with the *CAD* sketch of it and other images of the solution. How the edge detector works on a low level is beyond the scope of this thesis, but in LabVIEW the *IMAQ Find Edge VI* finds the straight edges by inputting an image source and a ROI for the edge detector to search within. This *VI* outputs the  $x$  and  $y$  pixel value.

### 3.2.2 The electrical circuit

In this section the final electrical circuit as well as a few suggested circuits are discussed and described. Benefits and weaknesses will be mentioned and considered along the way to make sure the flow of the reading experience is maintained. Some of the aspects of the discussion will be elaborated in chapter 5.

#### Earlier suggested versions

First, a circuit was suggested which used an Arduino Micro to provide with the PWM signal and analog reading of the voltage. This meant that the Arduino and the LabVIEW program was coded separately, where the Arduino periodically prints the analog read values to the serial port using the *Serial.print*-command. The LabVIEW program would then have to match the period with the reading of these values. This caused an "*iteration delay*" which is discussed in chapter 5.5.

Inspired by findings in Liu et al. (2009), where the duty cycle had an upper limit of 40% and a base frequency of about 100Hz, the circuit was designed by those same premises, mainly because the corresponding  $V_{R_k}$  was supposed to have sufficient settling time when the PWM signal was on/high. This turned out not to be applicable for this thesis, as the settling time was much shorter than 1ms. However, the circuit was designed to accommodate the low duty cycle and was therefore supplied with 30V voltage from the power supply and a low value of 4Ω on the resistor in series with the wire, to produce the desired effect input to the actuator wire. The circuit had two differential amplifiers, measuring both  $V_{R_k}$  and  $V_{tot}$ . However, since the voltage drop over the Darlington transistor is constant at  $\sim 0.75V$  when it is fully

on. Thus, the voltage drop over  $V_{tot}$  is also constant, and consequently the last differential amplifier is redundant.

The second version was quite similar to the first, but had only one differential amplifier measuring the voltage drop  $V_{R_k}$ . Also, the Arduino Micro was changed to an Arduino Mega2560 because the Micro was not compatible with the *LabVIEW Makerhub*, *Digilent LINX* package for LabVIEW. The Arduino is then used as a Data Acquisition (DAQ) device, meaning that it is used only for reading and sending sensor values to LabVIEW and producing a PWM signal as an output. The computations are done on the computer running the LabVIEW program. However, because of the problem regarding measurement timing of the analog read, a simple first order active low pass filter with amplification was implemented at the output of the amplifier to smooth the PWM signal and maintain the range of the ADC in the Arduino.

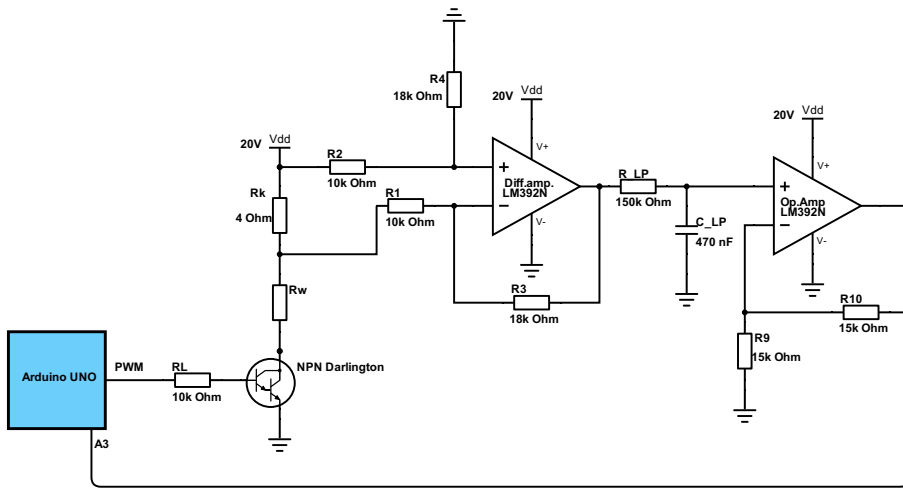
### Final solution

Lastly, the Arduino Mega2560 was replaced with an Arduino Uno due to the Mega2560 being defect, but is still being used as a DAQ device. The system is based on a PWM signal switching the Darlington transistor on and off. When the transistor is fully open/on a power supply is supplying the SMA actuator ( $R_W$ ) and a known resistor  $R_k = 4\Omega$  with  $20V$ . This is what causes the heating of the wire, and consequently the displacement in length. Although the current never exceeds  $280mA$ , according to the calculations, it is limited by the power supply to  $410mA$ . This limit acts like a "fail-safe" seeing as it is the maximum recommended current for the wire according to FLEXINOL<sup>®</sup> own homepage (Dynalloy, 2016). If the current exceeds this value it might damage the wire.

The actual voltage drop over the known resistor ( $V_{R_k}$ ) is inputted into a differential amplifier with the amplification factor  $A_{diff} = 1.8$  calculated by equation 2.7. This voltage drop is used to determine the current ( $I_C$ ) through the wire, such that the resistance of the actuator ( $R_W$ ) can be calculated by *Ohm's law*. From there, the amplified signal goes through a first order active low-pass filter (LP-filter) to avoid the issue regarding measurement timing with a PWM signal. This issue is discussed in detail in chapter 5.4. The cut off frequency ( $f_c$ ) of this LP-filter was calculated to  $2.25Hz$  by equation 2.10. Where  $R = 150k\Omega$  and  $C = 470nF$ , which yielded a sufficiently smooth signal. This filtered signal is subsequently fed in to a non-inverting operational amplifier with an amplification factor of  $A_{op} = 2$ , to scale up the measurement difference referenced to ground, calculated by equation 2.9, which is the amplification part of the active LP filter. This consequently increases the accuracy of the ADC in the Arduino. However, this filter introduced a few problems discovered later in the project period which are discussed in chapter 5.6.1.

The input resistors of the differential amplifier ( $R_1$  and  $R_2$ ) are selected to be much

larger than the known resistor  $R_k$  and the actuator  $R_W$ . This is because the main amount of the current is supposed to flow through the wire to create heat. Specifically, the resistors have the value  $R_1 = R_2 = 10k\Omega$ . The resistor  $R_L$  is a limiting resistor. This limits the current flow into the Darlington transistor and drives the PWM signal low when off.



**Figure 3.3:** Schematic of the final experimental circuit. Designed in *DigiKey* Digikey (2016).

### 3.3 Arduino and LabVIEW code

The front panel of the LabVIEW program, seen in appendix B, figure B.1 and B.2, serves as the Graphical User Interphase (GUI) of the system when the program is running. In this panel the user controls the system. Additionally, a few graphs are displayed which are monitoring the different calculations and measurements as well as the camera image showing the end of the actuator wire hanging in front of a ruler. This image is shown in figure D.1.

But, before starting up the LabVIEW program the camera must be configured. This is done by opening National Instruments - Measurements & Automation Explorer (NI MAX) and finding the camera in the subfolder “*Devices and Interfaces*”. Here the *exposure time* and *gain* are set to 3000 and 20, respectively, to get the desired brightness and frame rate of the image. This is however, dependent on the lighting conditions in the room where the setup is placed. Additionally, the dimension of the image is set to make it smaller and increase the potential frame rate and reduce the data size of the image. In this case the dimension is  $1408 \times 290$ , with an offset of  $472\text{pixels}$  from the bottom and  $470$  from the left. These numbers are chosen arbitrary, just to get a smaller image and to center the movement of the wire in the middle of the image. A full scale image or a smaller one would be just as good since the edge detector only looks for an edge within the ROI. This was done by measuring a known length of  $50\text{mm}$  with the ruler in the image and find the same length in *pixels*. It was measured  $361\text{pixels}$  in  $50\text{mm}$ . This means that there was about  $7.22\text{pixels}/\text{mm}$ . Although this is a crude method for finding the length with an image, it served its purpose well and gave a sufficiently accurate measurement.

Initially, the LabVIEW program also needs a few settings to be configured. These include; the file path for the saving of data, as well as for reading of the dataset on which the estimates are based, and the COM port connected to the Arduino. Additionally, a ROI must be defined in the image. Within this ROI is the workspace for the edge detector that produces the parameter called the *actual displacement* ( $D_S$ ) in this thesis. The value is found by calculating the mean between the start and the end point of the detected edge along the x-axis (see image D.1).

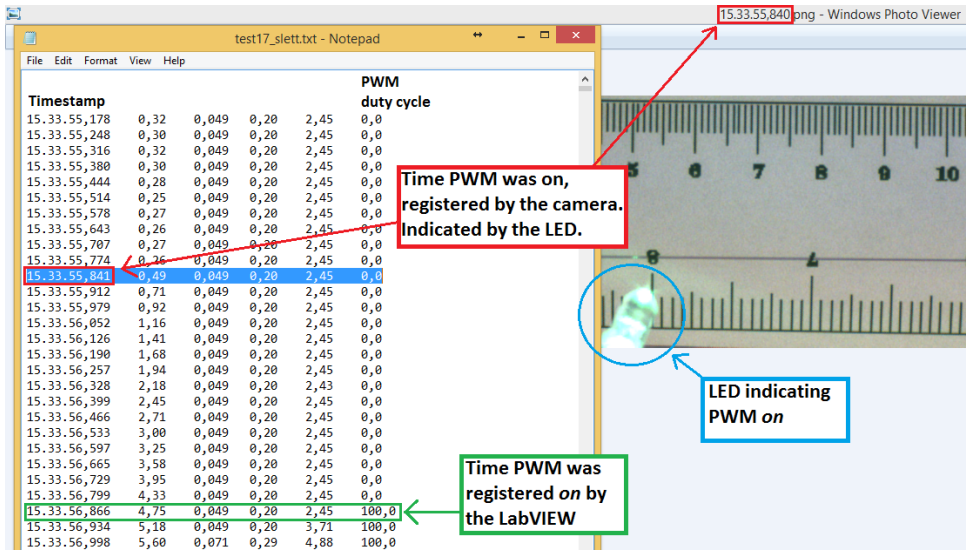
#### Earlier suggested versions

All the aforementioned settings apply to all the solutions to some extent. The only change is that at one point the program had the possibility of saving the images shown by the camera. This was used to detect the time delay from when the PWM signal was set *on* and the time it was registered in the LabVIEW code. Effectively, the time used to send and receive the signal to turn *on* or change the duty cycle of the PWM signal.



The main difference between the versions of the LabVIEW programs, apart from the possibility of saving the images from the camera, is the communication between the Arduino and the LabVIEW code. At first, the two were coded separately. The communication was done by using the VISA libraries in LabVIEW for printing to the serial port via LabVIEW and through the *Serial.print()*-command in the Arduino code. This means that the LabVIEW program was timing the receiving of data with the sending from the Arduino with no synchronization. As shown in image 3.4 the delay is 1.025sec, which is unacceptable. Therefore the LabVIEW and Arduino code was changed to neglect this delay. A solution to this was the Diligent LINX - LabVIEW Makerhub package for LabVIEW. The benefits and drawbacks of this as well as the time delay will be discussed further in chapter 5.5.

When the second solution was developed, the idea was to make a soft PWM to time the measurement of the analog read when the PWM signal was high. However, due to the reduced functionality when implementing LINX and the high frequency needed for the PWM signal, this proved to be difficult. The LINX package for LabVIEW only allow a few functions, where *analog read*, *analog write*, *digital read* and *digital write* are some of the most important in this case. When the soft PWM was implemented, the Arduino could not set the digital output pin high, read the analog input pin, then set the digital output pin low fast enough. The attempted code only caused the actuator to make a jerking motion because the frequency was too low.



**Figure 3.4:** Image of the saved data (left) and the first saved image of the instant the LED was turned on (right), indicating the PWM signal was on.

### Final solution

There was made two different editions of the final solution. The first one saves all the measured and calculated parameters used for finding the polynomial functions for the different estimates. The second edition is the version used to control the SMA actuator. The only difference between the two, is the fact that they save different parameters. The first one is based on a step response. The later edition is the one used for controlling the actuator.

In the front panel of the LabVIEW program, the user can turn on and off the PID controller via a toggle switch while the program is running. When the PID is on, the set point can be changed at any point in time. When the PID is off the user can control the duty cycle from the Arduino directly. There are four different buttons on the front panel, one is controlling the saving of the data (*on/off*), a second is calibrating the zero point for the measurement of displacement by the camera (*latch when pressed*), the third initializes the PID controller by canceling the built up integral. The last button stops the whole program.

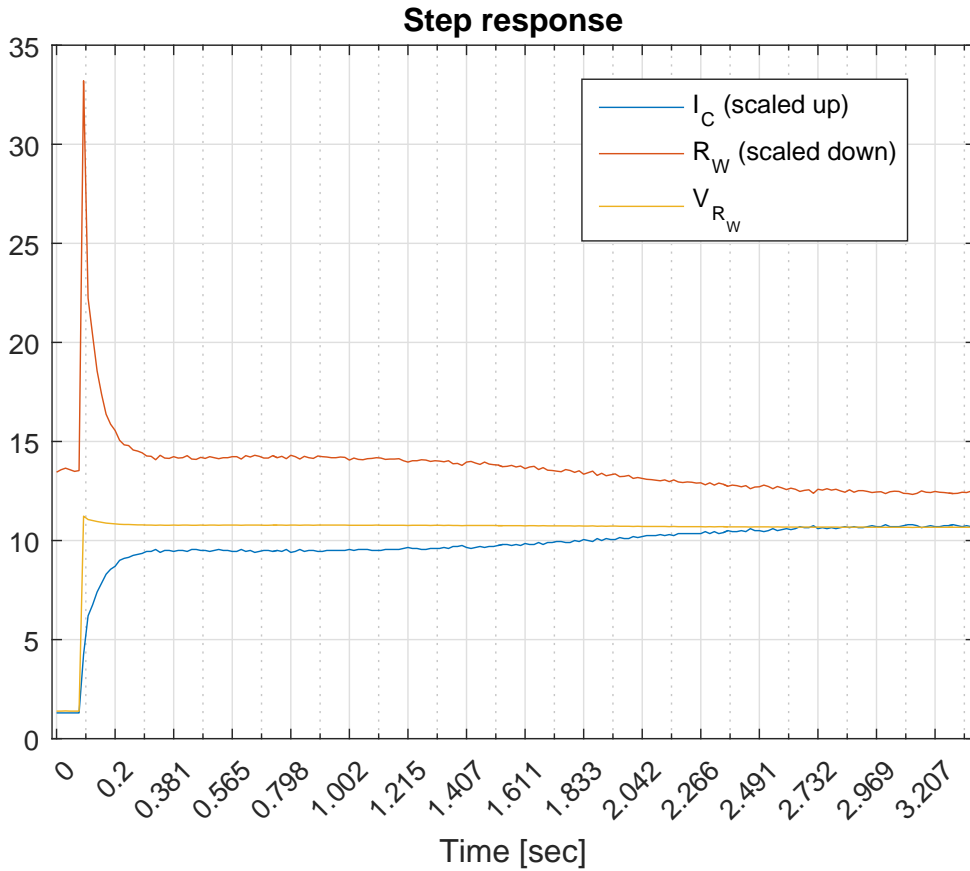
One occurring problem was that the  $R_W$  value spiked at the moment the duty cycle changed to rapidly. Therefore, it was implemented a low-pass filter to reduce these rapid changes. Figure 3.5 illustrates the cause of the spike. When a step is inputted on the wire the voltage  $V_{R_W}$  reacts instantly, where the dynamics of the current

is slower. This may be because  $I_C$  is dependent on the polynomial fitted function, which will take more time depending on the order of the polynomial. This will be discussed further in chapter 5.2.

One temporary solution to the problem was to implement a LP-filter to smooth the rapid changes in duty cycle. This was tested in chapter 4.1.4. By comparing figure 3.5 and 3.6, one can see that the spike is attenuated significantly. Additionally, the duty cycle was limited to a range of 8 - 80%. The lower limit was because when the value got too low, the estimates diverged as a result of the polynomial fitted functions which are dependent on the duty cycle and the bias of the amplification.

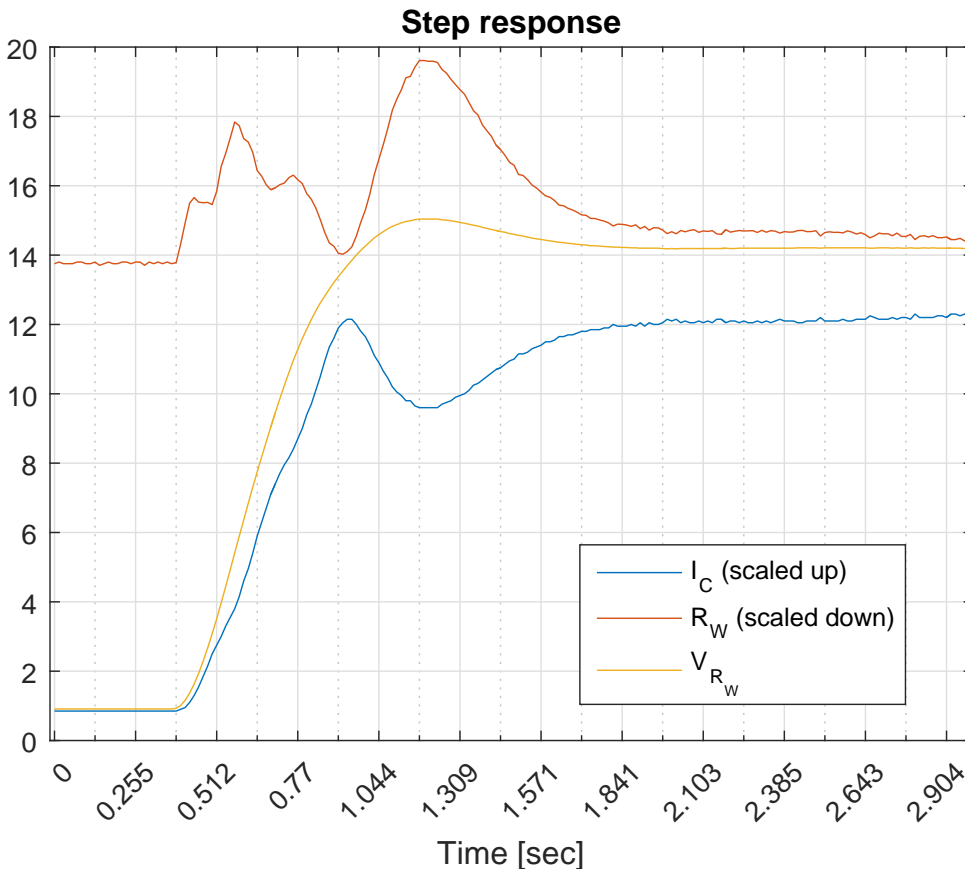
The upper limit is partially set because of the initial impression that the duty cycle was supposed to be as low as possible, and were never readjusted to the max value. The other part of the reason was that the filter allows for some overshoot, which means it can exceed the upper limit for a small moment, but if the value were to exceed a value of 100% (= 255 in *register value*) it would cause the register value to overflow and start back at 0% duty cycle. However, the upper limit could advantageously be set higher. These topics are discussed further in chapter 5.6 as well.

This plot shows the cause of the spike in  $R_W$ . Since  $R_W = \frac{V_{R_W}}{I_C}$  is will become large when  $V_{R_W}$  is large and  $I_C$  is low.



**Figure 3.5:**  $V_{R_W}$ ,  $\tilde{I}_C$  and  $\tilde{R}_W$  plotted together to illustrate the cause of the spike in  $R_W$ . Some of the values are scaled for easier comparison ( $\tilde{I}_C = 50 \cdot I_C$  and  $\tilde{R}_W = R_W/4$ ).

The maximum value of  $R_W$  is now attenuated from  $R_W = 132.8\Omega$  in figure 3.5 to  $R_W = 78.5\Omega$ . However, these are not real values. They are results of a poor measurement.



**Figure 3.6:**  $V_{R_W}$ ,  $\tilde{I}_C$  and  $\tilde{R}_W$  plotted together to illustrate the cause of the spike in  $R_W$ . Some of the values are scaled for easier comparison ( $\tilde{I}_C = 50 \cdot I_C$  and  $\tilde{R}_W = R_W/4$ ).

The behavior of the plots will be discussed in chapter 5.2.

### 3.4 Mathematical model

To be able to implement the inverse compensator from Liu et al. (2009) the mathematical model of the wire must be designed. The idea was to reproduce the findings in this article, such that the model could be verified and for saving time.

However, the model proved to be too complex and was therefore excluded from the solution of this project. More specifically, the implementation of the minor loops in the model was challenging, and consequently the tuning of the different parameters. It was difficult to pinpoint exactly where the problem was and the physical relation of the different parameters were difficult to find. The value of the parameters were similar, but the results were different, and eventually the model of the system was scrapped.

# Chapter 4

## Testing and results

Some of the tests conducted for this thesis are used for validation and evaluation of the final solution. They are used to find whether or not the solution worked, and to assess the performance. Others however, are used to make functions for estimating the desired parameters. All the tests were conducted in the same environments. The room temperature was about 20°C and the lighting conditions were good with the opportunity of blending out the sun if needed.

Before conduction a test, there are a few things which needs to clarified, such that the intentions and purpose of the tests are clear:

1. What is the objective of the test?
2. Why conduct this test?
3. What was expected to happen?
4. What actually happened?
5. How can these result be applied?

These questions will be answered sequentially for each test.

Following in this chapter are explanations of the different conducted tests and their results. The results will also be interpreted and discussed with respect to expectations, relevance and reliability, and some consequences will be further discussed throughout chapter 5.

Firstly, the open loop test will be presented. A few graphs and tables will supplement the explanation, giving a sufficient amount of information about the tests.

Secondly, the performance of the tracking ability of the PID controller will be assessed. Similar to the open loop test, a few graphs will be presented to give the reader the ability to make own assessments regarding the performance.

## 4.1 Open loop test

To get approximately the same tracking results from the controller as in the article by Liu et al. (2009), without implementing the hysteresis model for the SMA actuator, a polynomial function of the displacement dependent on the resistance  $R_W$  was calculated based on measurements made in an open loop test. This test was conducted by inputting a step with the PWM signal and logging the corresponding calculated values such as:  $V_{R_k}$ ,  $I_C$ ,  $V_{R_W}$ ,  $R_W$  and  $D_S$ . Figure 4.6, 4.7, 4.8, and 4.9 shows the step response of the system before and after filtering the duty cycle.

All these values are calculated based on the *analog read* from the Arduino and can be called estimates, but in this thesis they are referred to as calculations.

### 4.1.1 Estimate of $V_{tot}$

The objective of this test is to get an accurate estimate of the voltage drop over both  $V_{R_W}$  and  $V_{R_k}$  called  $V_{tot}$  without measuring it.

After filtering the PWM signal to circumvent the issue of timing the analog read with the PWM signal, the voltage drop over both the actuator wire ( $V_{R_W}$ ) and the known resistor ( $V_{R_k}$ ), meaning  $V_{tot}$ , was observed to be varying. Since this was detected late in the project period, the voltage drop was therefore measured with a multimeter and fitted with a polynomial function dependent on the PWM duty cycle (DC) from the Arduino. Table 4.1 shows the measurements.

The measured results was expected to linearly increase proportionally to the duty cycle. The “*Expected [V]*” column in table 4.1 lists the expected value and is calculated by multiplying the 20V supply voltage with the duty cycle percentage.  $Expected = \frac{DC}{100} \cdot V_S$ , where  $V_S$  is the supply voltage.

Table 4.1 shows that the expected values are slightly of the actual measured values, but the estimate calculated from the fitted polynomial function (equation 4.1) is quite similar to the measurements.



DC [%]	$V_{tot}$		
	Expected [V]	Measured [V]	Estimated [V]
5	1,0	1,01	0,98
10	2,0	1,90	1,94
15	3,0	2,88	2,89
20	4,0	3,85	3,85
25	5,0	4,82	4,80
30	6,0	5,79	5,75
35	7,0	6,69	6,70
40	8,0	7,65	7,66
45	9,0	8,62	8,61
50	10,0	9,52	9,56
55	11,0	10,49	10,50
60	12,0	11,45	11,45
65	13,0	12,42	12,40
70	14,0	13,39	13,35
75	15,0	14,28	14,30
80	16,0	15,25	15,25

**Table 4.1:** The measurement of  $V_{tot}$  is done with a multimeter and the calculation is done in the LabVIEW code. The estimate is the output of the fitted polynomial function.

The corresponding polynomial function based on this table is of the third degree and is expressed as follows:

$$V_{tot} = 1.55 \cdot 10^{-7} DC^3 - 2.85 \cdot 10^{-5} DC^2 + 0.19 DC + 0.026 \quad (4.1)$$

Where  $DC$  is the duty cycle. This function was then used to estimate the voltage drop  $V_{tot}$  and subsequently calculating  $V_{R_W}$ .

Ideally, the value would be measured continuously in the same way as the voltage drop over the known resistor ( $V_{R_k}$ ), but since ( $V_{tot}$ ) was thought to be constant up until late in the project period a temporary solution was implemented because there was no time to get another operational amplifier. This will be discussed further in chapter 5.6. However, function 4.1 was implemented in the LabVIEW code and gave sufficiently accurate calculated estimates of the voltage drop.

### 4.1.2 Amplification compensator

The objective of this test is, similar to the test in chapter 4.1.1, to get an accurate estimate of the amplification based on fitting a polynomial function to actual measurements to compensate for the biased amplification illustrated in figure 4.1.

This test is conducted to compensate for a bias caused by the capacitor in the low-pass filter in the circuit. Ideally, the problem should have been rectified in the circuit, but because the problem was detected at a late stage in the project period and the needed components were taken, a compensator was designed.

The amplification value was initially expected to be constant at the value determined by the configuration of the resistors. For the differential amplification this value is 1.8 and for the operational amplifier this value is 2. This yields a total amplification of 3.6. However, since the values of the calculated  $V_{R_k}$  was observed to vary before the compensator was implemented the amplification was expected to be a decreasing function which would converge towards the actual amplification of 3.6.

The calculations shows that the amplification is not constant, but a decreasing function of the duty cycle, illustrated in figure 4.1. This is because of the stored voltage in the low-pass filter capacitor ( $C_{LP}$ ). How this happens and the consequences which follows is discussed in chapter 5.6.1. Table 4.2 shows that when the input voltage is low the registered *raw data* from the Arduino is amplified the most. At the moment the duty cycle is 5% the measured voltage drop over  $V_{R_k}$  is  $0.07V$  and the *raw data* is  $1.49V$ . This means that the input signal has an amplification of  $\frac{1.49V}{0.07V} = 21.29$ . Figure 4.1 illustrates the change in amplification dependent on the duty cycle of the PWM signal. The *Estimated* column under *Amplification* in table 4.2 is the values calculated from equation 4.2.

This estimated amplification value is used to divide the *Raw data* from the Arduino to get an estimation of  $V_{R_k}$ , thus compensating for the biased amplification in the circuit.

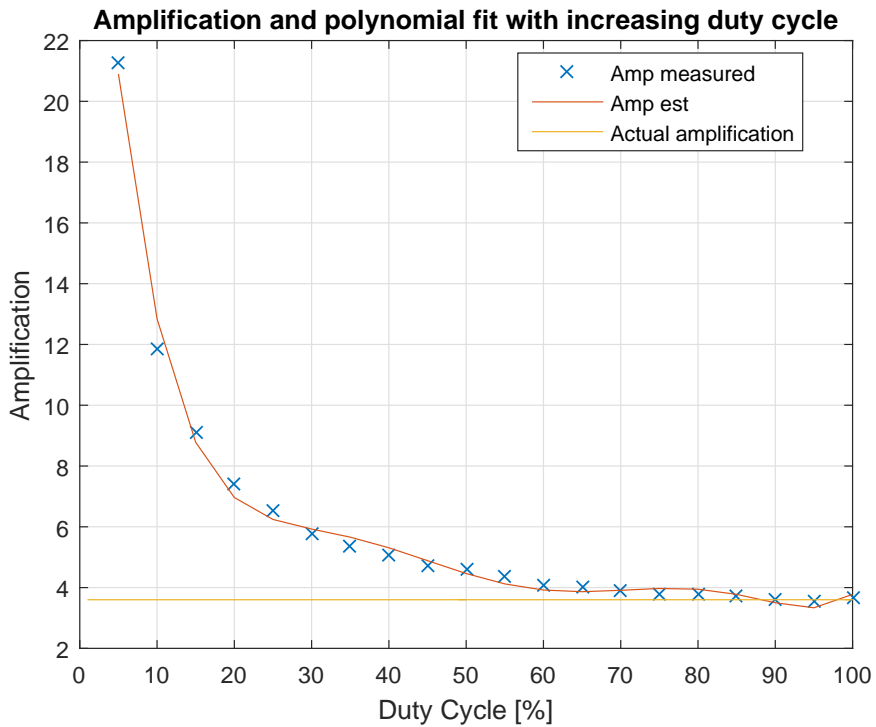
DC[%]	$V_{R_k}$		Amplification	
	Measured[V]	Raw data[V]	Measured $A_E$	Estimated $A_M$
5	0,07	1,49	21,29	21,10
10	0,14	1,66	11,86	12,47
15	0,2	1,82	9,10	8,67
20	0,27	2,00	7,41	7,18
25	0,33	2,16	6,55	6,54
30	0,40	2,32	5,8	6,07
35	0,46	2,46	5,35	5,56
40	0,52	2,63	5,06	5,03
45	0,59	2,79	4,73	4,60
50	0,64	2,94	4,59	4,36
55	0,72	3,13	4,35	4,28
60	0,91	3,70	4,07	4,26
65	0,99	3,95	3,99	4,16
70	1,07	4,17	3,90	3,89
75	1,15	4,37	3,80	3,58
80	1,22	4,59	3,76	3,85
85	1,29	4,77	3,70	3,78
90	1,37	4,95	3,61	3,50
95	1,44	5+ (5,13)	(3,56)	3,33
100	1,49	5+ (5,42)	(3,63)	3,77

**Table 4.2:** The measurement of  $V_{R_k}$  is done with a multimeter and the Raw data is directly from the Arduino, displayed in the LabVIEW code. The values in the parenthesis are based of measurements with multimeter because the value exceeded the upper value for the Arduino ADC.

In table 4.2 the amplification measurement is calculated by dividing the raw data, read by the Arduino, with the measured voltage ( $A_M = \frac{Raw\ data}{Measured}$ ). The estimate is the output of the polynomial fitted function, shown in equation 4.2 at the given duty cycle input. The corresponding equation of the polynomial function of the amplification is of the sixth degree and is expressed as follows:

$$\begin{aligned}
 A_E = & 4.94 \cdot 10^{-9} DC^6 - 1.41 \cdot 10^{-6} DC^5 + 1.61 \cdot 10^{-4} DC^4 \\
 & - 9.37 \cdot 10^{-3} DC^3 + 0.29 DC^2 - 4.74 DC + 38.59
 \end{aligned} \tag{4.2}$$

The following graph is based on equation 4.2 and measurements shown in table 4.2.



**Figure 4.1:** Measured and calculated amplification with a polynomial fitted function.

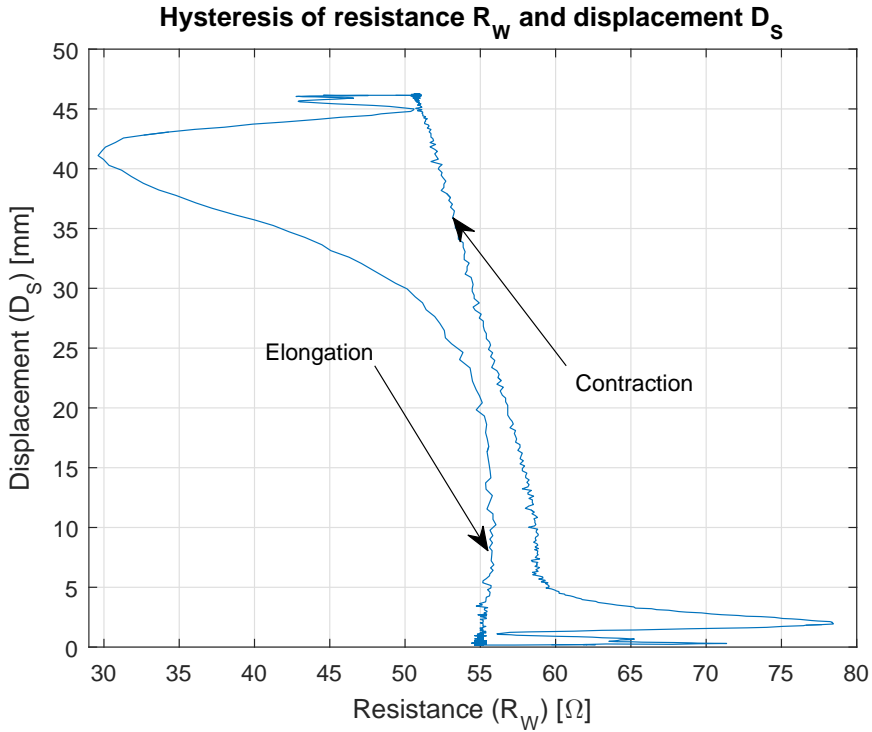
The amplification can be observed to be much larger for low values in duty cycle than for higher values. However, the amplification converges towards the actual value of  $1.8 \cdot 2 = 3.6$  as the duty cycle increases. How the problem occurs and possible solutions to it is discussed in chapter 5.6.1.

### 4.1.3 Estimate of the displacement

The objective of this test is to make an estimate of the displacement based on the resistance ( $R_W$ ) registered in the wire, utilizing the self-sensing ability. The aim is to fit a polynomial function to a dataset of the resistance and the actual displacement from a step response test run prior to the main program, similar to the other estimate tests.

To get an estimate of the displacement without using external sensors or implementing a model of the system, the relation between  $R_W$  and  $D_S$  must be expressed with a fitted function based on earlier performance. In this case a polynomial function was chosen.

Based on work done by others in the field (Liu et al. (2009) and Dutta and Ghorbel (2005)) the relation between the two parameters of the actuator was expected to be close to linear in the contraction phase. In this case the readings would not make sense if they were registered in the elongation phase, because the PWM signal would be low (close to 8% duty cycle) and the circuit would essentially be “turned off”. In figure 4.2 the full hysteresis characteristics of the wire is plotted. It can be observed that at the start of both the contraction and elongation phase of the displacement there is a big deviation. This is because of the value of the calculated current ( $I_C$ ) has a drop in its value as the displacement of the wire changes. Thus, causing the corresponding  $R_W$  to also perturb. Therefore the polynomial function was fitted to the area of the contraction in which were close to linear. This is discussed in chapter 5.5.1.



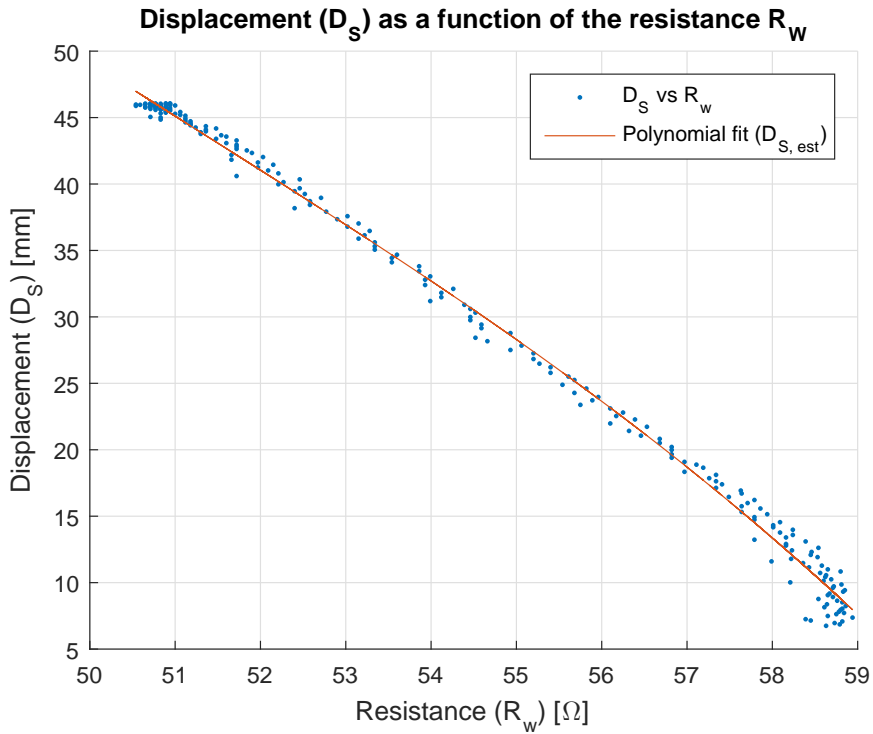
**Figure 4.2:** The hysteresis characteristics of the wire.

Figure 4.3 shows the fitted polynomial function compared to the measurements of the hysteresis in the contraction phase. It can be seen that the graph seems close to linear. However, the function was fitted to a polynomial of the third degree to increase the accuracy and still keep the function simple. Equation 4.3 shows how the relation is expressed.

The fitted polynomial function is of the third degree and is expressed as follows:

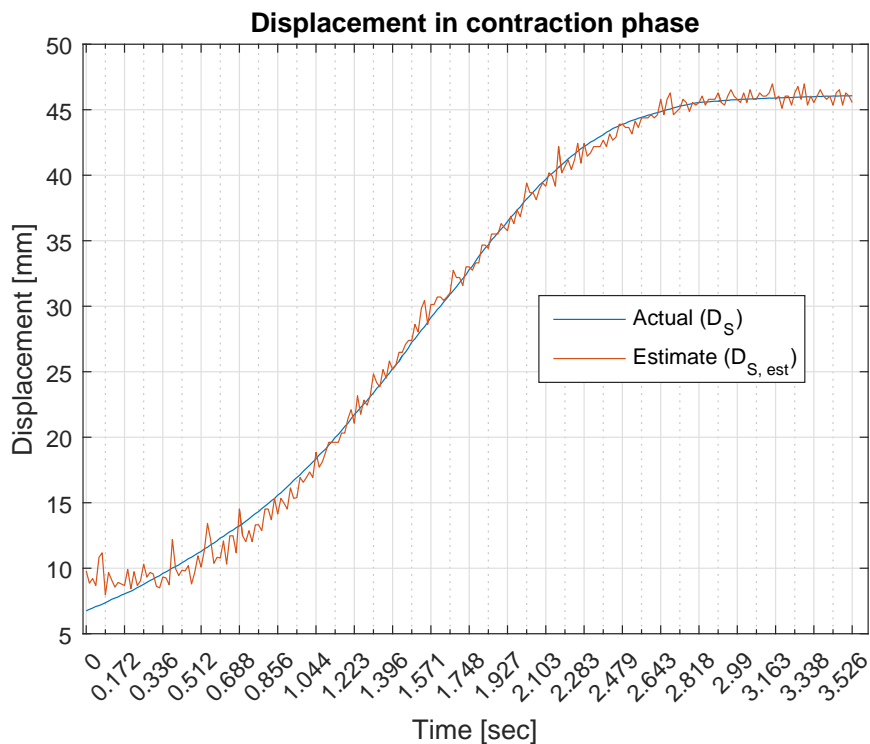
$$D_{S,est} = 0.015 \cdot R_W^3 - 2.47 \cdot R_W^2 + 133.24 \cdot R_W - 2279.4 \quad (4.3)$$

These results can be used to estimate the displacement of the actuator. Essentially, fulfilling the task of using the self-sensing ability of the actuator. Figure 4.4 and 4.5 shows the accuracy of the estimated displacement relative to the actual displacement. However, these are results based on a step response where the contraction goes from resting state (fully elongated) to fully contracted. Therefore, the estimate might not work as good for different changes in displacement.



**Figure 4.3:** A polynomial fitted function of the  $D_{S,est}$  with respect to the resistance ( $R_W$ ).

The estimate can be observed to follow the actual displacement quite good. Even though the accuracy of the estimate deteriorates as the values goes under  $10mm$ , the performance is satisfying compared to the expectations. Figure 4.5 illustrates the accuracy with a bar graph of the error between  $D_S$  and  $D_{S,est}$ .

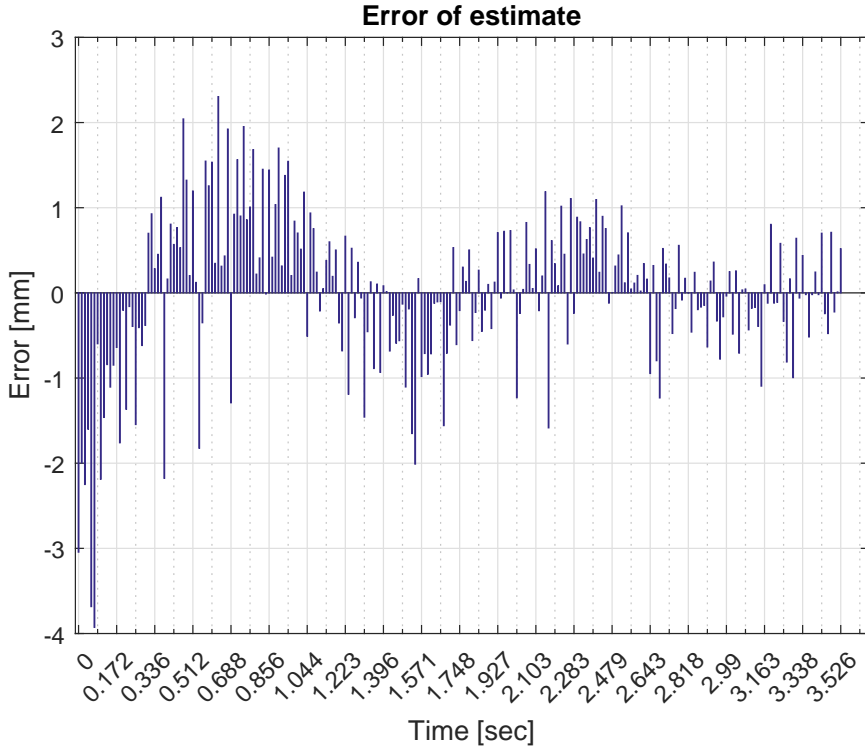


**Figure 4.4:** Step response of the actual displacement ( $D_S$ ) compared to the estimated displacement ( $D_{S,est}$ ).

The estimate error is in the range of 2 - 4% of the actual displacement ( $D_S$ ) shown in figure 4.5 - hence sufficiently accurate for such applications as gripper positioning with tactile feedback.



This plot shows the when the contraction start the error is at approximately 4% and decreases as the displacement increases.



**Figure 4.5:** Bar graph of the error between  $D_S$  and  $D_{S,est}$ .

In chapter 4.2 the performance of the estimate is tested and briefly assessed.

#### 4.1.4 Step response

The objective of this test is to gather the data needed to produce the estimates. The idea is to conduct this test before running the main program, such that the collected data is used at the start of the program, initiating the polynomial fitted functions.

The test was conducted by pressing the “*Save DATA*” button in the front panel of the LabVIEW program while the wire is fully elongated, and then driving the duty cycle from the minimum limit (8%) and up to the maximum limit (80%), essentially inputting a step to the system. While the temperature of the actuator wire rises the parameters change proportionally to the contraction and consequently saved. The “*Save DATA*” button is switched off and the data is stored.

The step response in this thesis is used to make the estimates by fitting a polynomial function to the data logged from the tests. Parameters such as  $V_{R_k}$ ,  $V_{R_W}$ ,  $I_C$ ,  $R_W$  and  $D_S$  are calculated based on the measured amplified voltage drop over the known resistor ( $V_{R_k}$  actual). This is done 50 times per second and stored in a *.txt* document.

All the parameters was expected to be instantaneous along with the step function and then converge towards a steady state value as the resistance dropped as a result of the temperature change and the actuator wire contracted to its maximum.

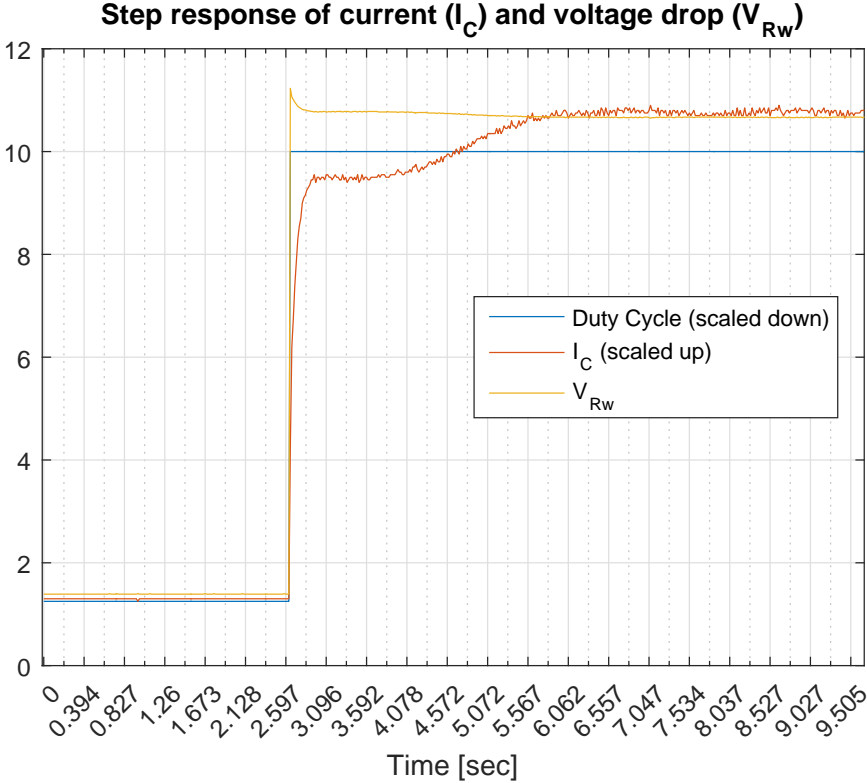
$R_W$  was observed to have a spike in the calculated value as the step was inputted. This is illustrated in figure 4.6 and 4.7. The effect of the spike in resistance value was amplified through the program and resulted in a even bigger spike in the estimated displacement when implemented. Therefore, the duty cycle input was filtered through a first order LP filter in the LabVIEW code, effectively implementing a reference model for the change in duty cycle. This resulted in a smaller and slower spike in the calculated  $R_W$  value, which can be seen in figure 4.8 and 4.9. Additionally, the displacement estimate became slightly better with respect to large perturbations. Possible reasons why this spike appear and ways to solve it will be discussed in chapter 5.2.

These results can be used for estimating the parameters needed for controlling the wire. Since there was no model of the system applied, this open loop step response was crucial for the data collection and consequently to approximate the different characteristics.

Following are the plots for the filtered and unfiltered step responses, respectively.

**Unfiltered duty cycle**

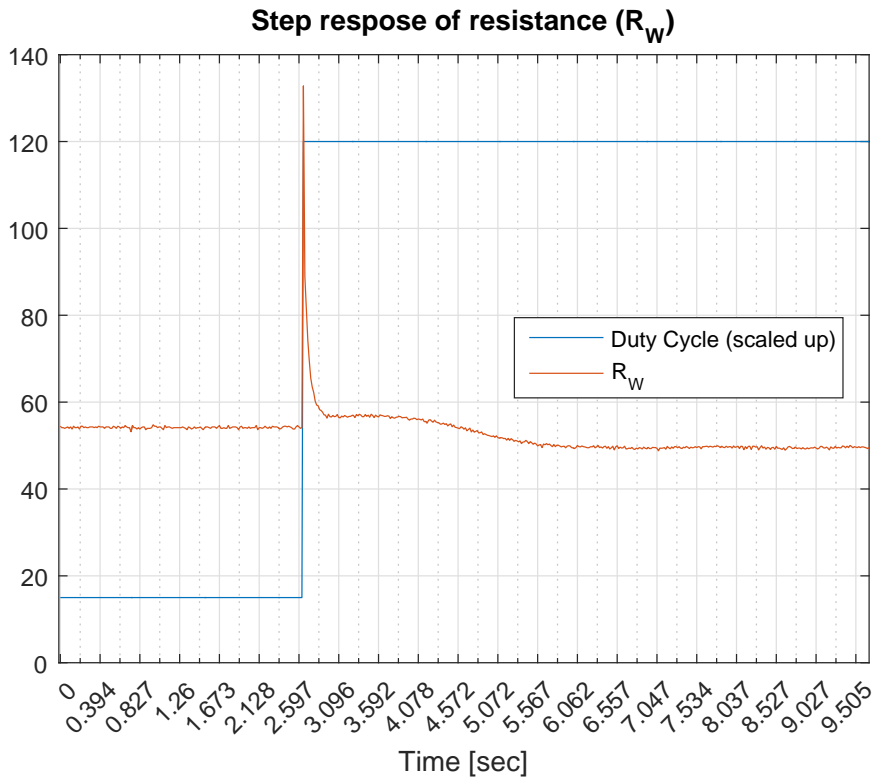
The following plots are of the step response with the unfiltered duty cycle input. The spike of  $R_W$  is illustrated in figure 4.7.  $V_{R_W}$  and  $I_C$  was calculated in the LabVIEW program based on the raw data from the *analog read*.



**Figure 4.6:** Step response of  $V_{R_W}$  and  $I_C$ . The values have been scaled to compare responsiveness. ( $\tilde{DC} = DC/4$  and  $\tilde{I}_C = 50 \cdot I_C$ )

$I_C$  is observed to change slower than  $V_{R_W}$ , thus making the relation between them large and consequently making  $R_W = \frac{V_{R_W}}{I_C}$  large.

This graph illustrates the consequences of the difference in responsiveness, which results in the spike in  $R_W$ .

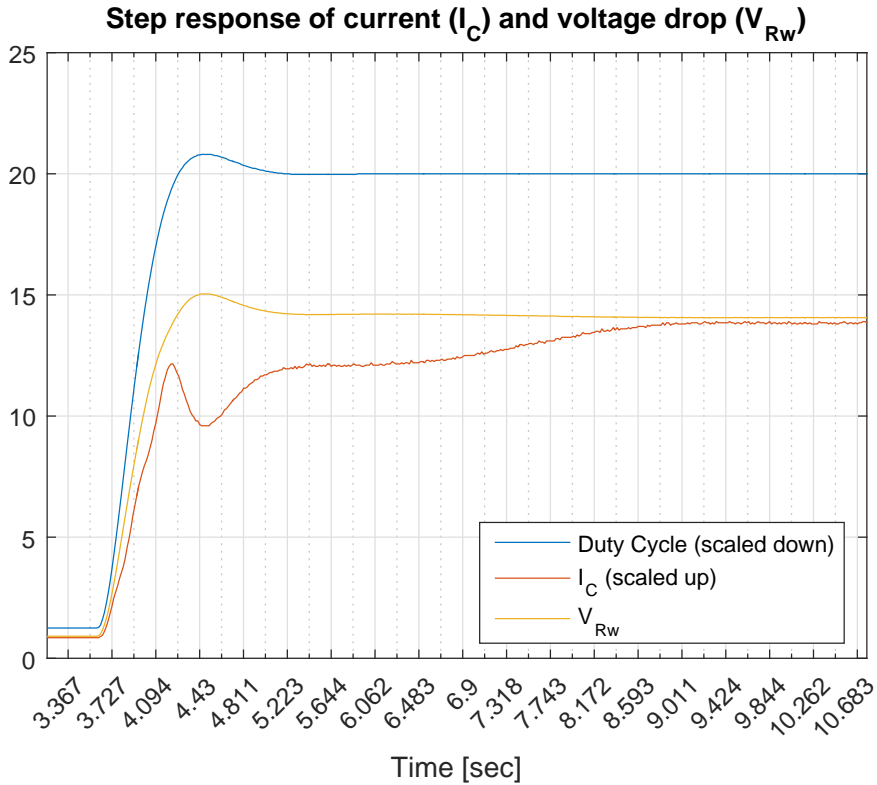


**Figure 4.7:** Step response of  $R_W$ . The values have been scaled to compare responsiveness ( $\tilde{DC} = 3 \cdot DC$ ).

This is however, not the actual behavior of the resistance in the wire, but a result of the poor measurement. This is discussed in chapter 5.2.

### Filtered duty cycle

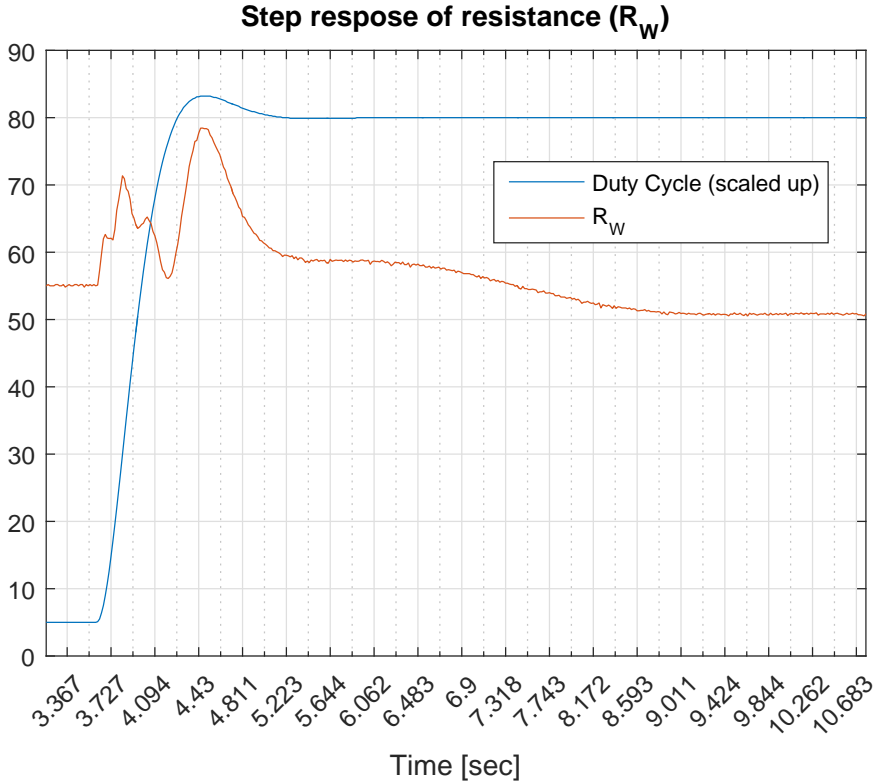
The following plots are of the step response with the filtered duty cycle input. The dampened spike of  $R_W$  is illustrated in figure 4.9. As the test done for an unfiltered duty cycle,  $V_{R_W}$  and  $I_C$  was calculated in the LabVIEW program based on the raw data from the *analog read*.



**Figure 4.8:** Step response of  $V_{R_W}$  and  $I_C$ . The values have been scaled to compare responsiveness. ( $\tilde{DC} = DC/4$  and  $\tilde{I}_C = 50 \cdot I_C$ )

The slight "dip" in value for the current ( $I_C$ ) is a result of the contraction of the wire and consequently the change in resistance  $R_W$ .

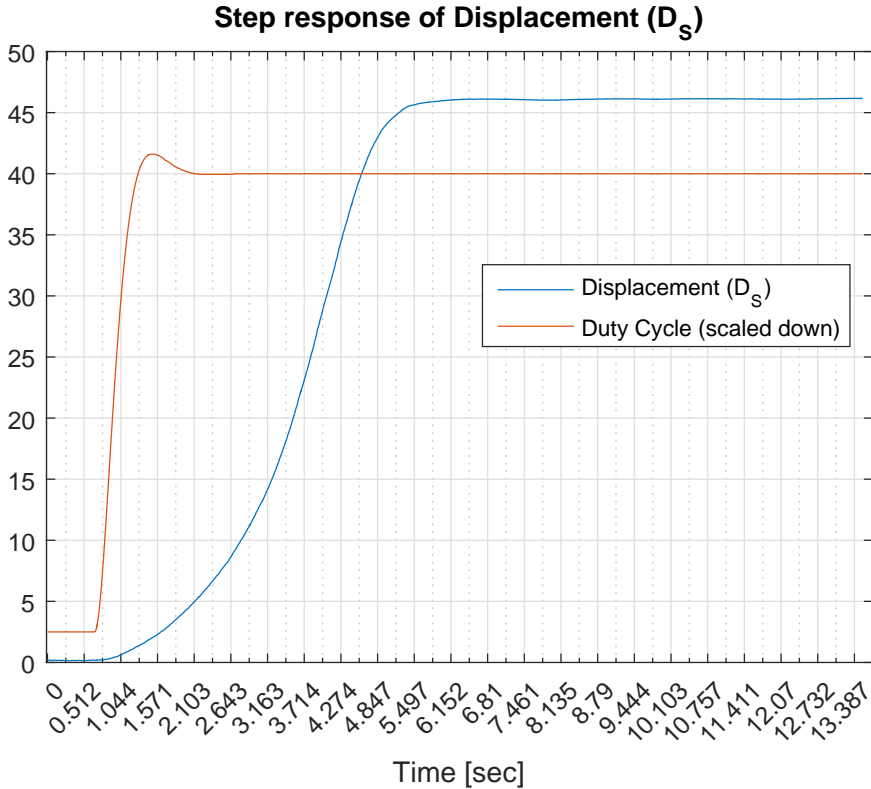
This graph shows the spike attenuated by filtering the input duty cycle. The first small spike comes from the difference in responsiveness, and the "dip" comes from the start of the contraction similar to  $I_C$  in figure 4.8.



**Figure 4.9:** Step response of  $R_W$ . The values have been scaled to compare responsiveness ( $\bar{DC} = 3 \cdot DC$ ).

Like in figure 4.7 this is not the actual physical behavior of the resistance value, but a result of the different dynamics of the parameters and the displacement of the wire. This will be discussed further in the chapter 5.2.

Figure 4.10 shows the step response of the actual displacement ( $D_S$ ) with the filtered duty cycle input.



**Figure 4.10:** Step response of  $D_S$ . The values of the duty cycle have been scaled to compare responsiveness ( $\tilde{DC} = DC/2$ ).

The response can be observed to be relatively fast. However, it could have been faster if the effect inputted to the wire were closer to the recommended approximated maximum values.

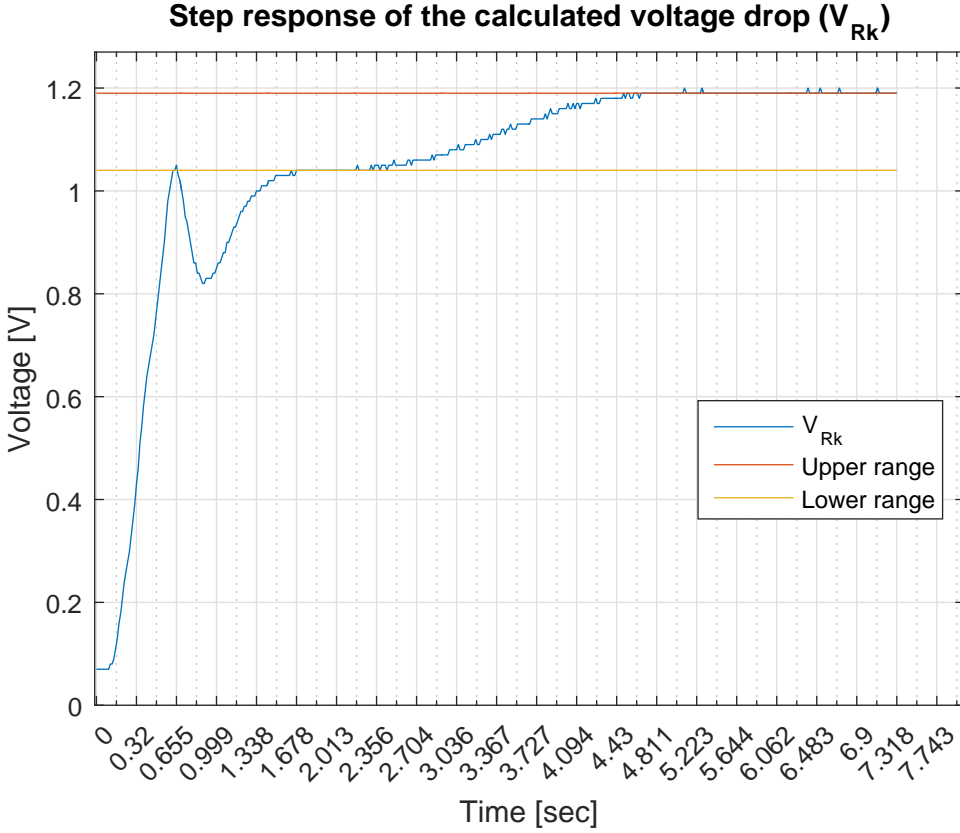
The corresponding transfer function was approximated in MATLAB based on the values plotted in figure 4.10. Equation 4.4 expresses the transfer function ( $H_{D_S}$ ):

$$H_{D_S}(s) = \frac{0,1156}{s^2 + 0,556s + 0,21} \quad (4.4)$$

This is a second order transfer function with two poles and no zeros. The function dependent on time can be found by inverse *Laplace transforming*  $H_{D_S}$ . However, this is not done due to its relevancy compared to a function of  $D_S$  dependent on  $R_W$ .

**Physical values of  $V_{R_k}$** 

Figure 4.11 illustrates the range of the voltage drop the analog read registers. The upper range is at 1.19 V and the lower range is at 1.04 V.



**Figure 4.11:** Plot of the actual range of the voltage drop ( $V_{R_k}$ ).

If the same values were to be calculated from  $R_W$  when the wire is heated and cooled, and  $R_k = 4\Omega$  by calculating the following equation:

$$V_h = \frac{R_k}{R_{W,h}} \cdot \tilde{V}_{tot} \quad (4.5a)$$

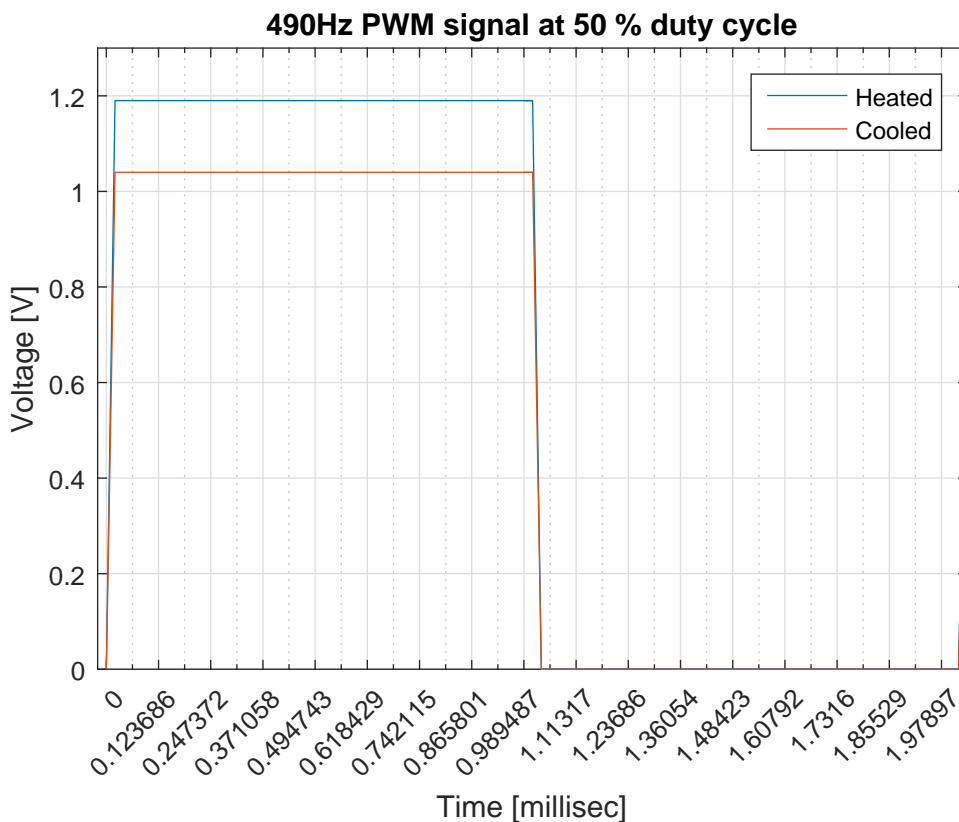
$$V_c = \frac{R_k}{R_{W,c}} \cdot \tilde{V}_{tot} \quad (4.5b)$$

Where the indexed  $h$  and  $c$  denotes the resistance value ( $R_W$ ) when the wire is heated and cooled, respectively, and  $\tilde{V}_{tot}$  denotes the constant voltage drop over  $V_{R_W}$  and  $V_{R_k}$  when the reading is timed with the PWM signal.



It would be:  $V_h = \frac{4\Omega}{50\Omega} \cdot 19.25V = 1.54V$  for the upper range, and  $V_c = \frac{4\Omega}{55\Omega} \cdot 19.25V = 1.4V$ .

Consequently, the same range plotted as the oscilloscope read it is illustrated in figure 4.12.



**Figure 4.12:** The PWM signal changing when the wire heats up.

This plot shows that how the voltage drop  $V_{R_k}$  changes when the PWM signal is high.

## 4.2 PID test

The objective of the following tests is to assess the tracking ability of the PID controller implemented in the system by observing the behavior of the displacement ( $D_S$ ). Additionally, the change of accuracy and response when the feedback is changed to something more accurate by using the actual displacement ( $D_S$ ) measured by the camera as feedback instead of the estimated displacement ( $D_{S,est}$ ), will also be assessed.

The tests were carried out by giving the system a set point in displacement that would follow a particular sequence such that the same input sequence could be repeated later for a different feedback. Then the tracking ability would be observed and the parameters indicating the performance of the controller were saved. Firstly, the estimated displacement ( $D_{S,est}$ ) would be used as the feedback parameter. Then, the same input sequence for the set point is inputted to the system with  $D_S$  as feedback. The input sequence was controlled by hand, and therefore the time intervals of the steps are not consistent.

This test was conducted to validate if  $D_{S,est}$  is good enough to make the  $D_S$  to get to the set point, as well as testing if the PID controller can bring the displacement to its set point even though the estimate is noisy.

The performance of the estimate was expected to have some perturbations and be sensitive to changes in DC because of the performance of the estimates, but still be accurate enough to bring  $D_S$  close to the set point. Therefore, the tracking capabilities were expected to be relatively poor compared to  $D_S$ .

However, as illustrated in figure 4.13 and 4.15, where  $D_{S,est}$  is used for feedback,  $D_S$  is following the set point to a certain extent with some noise and offset, with the exception at the start of the plot. Nevertheless, the performance of the controller was better than expected. Considering the poor performance of the estimate, the actual displacement tracked the set point with just a few millimeters off.

These results as they stand can be used as a “*proof of concept*”. Even though the system in this thesis is suboptimal with respect to performance and has room for improvement the end result still managed to behave and perform close to the intended task, which is to track the set point. This will be discussed further in chapter 5.1.

In chapter 4.2.1, which comprises “*PID-test1*” and “*PID-test2*”, the feedback is  $D_{S,est}$ , and in chapter 4.2.2, which comprises “*PID-test3*” and “*PID-test4*”, the feedback is changed to  $D_S$  for comparison for the different behavior.

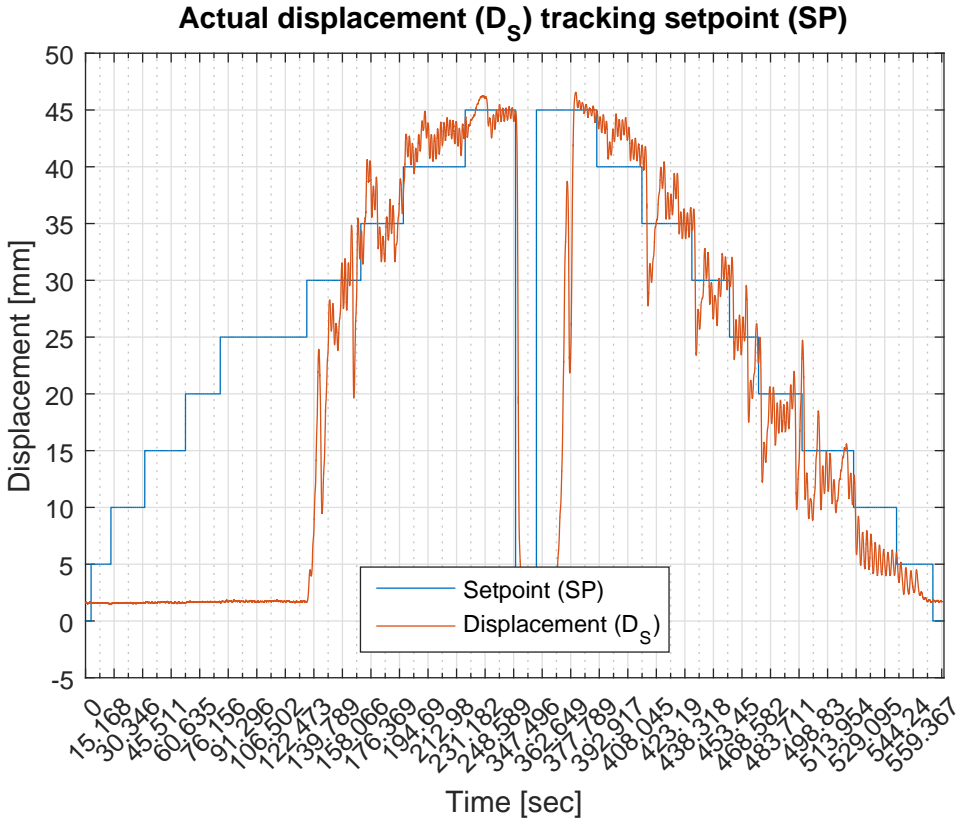
### 4.2.1 Estimate feedback

In the tests conducted in this section the estimated displacement ( $D_{S,est}$ ) was used as the feedback. There were performed different tests with different trimmed *PID*-parameters to see the change in tracking ability. The performance of the test is rated on the subjective evaluation by the author of this thesis.

In this test a stairs and a step function was inputted. The stairs function was steps of  $5mm$  and the step is from the minimum to maximum displacement ( $0 - 45mm$ ). In this way the sensitivity to large as well as medium changes in set point are tested. The next step would come if the system had settled within reasonable time. If  $D_S$  did not reach the set point within this time, it would increase to the next step.

**PID-test1: *Medium steps up, step down, then step up and medium steps down***"

In this test the PID parameters were:  $K_p = 4$ ,  $T_i = 40$  and  $T_d = 1$ . The graph shows the actual contraction ( $D_S$ ) following the set point poorly. The performance will be discussed in chapter 5.1.

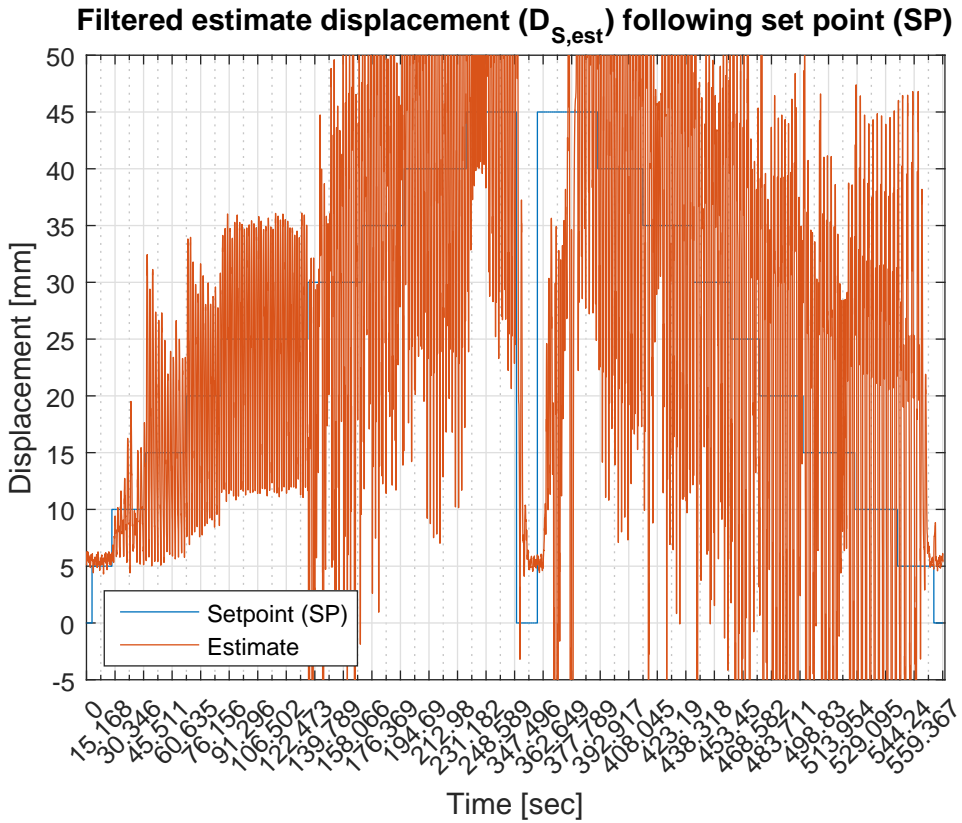


**Figure 4.13:** PID-test1:  $D_S$  tracking of set point with  $D_{S,est}$  as feedback.

It took 118.5seconds (1minute and 59seconds) before  $D_S$  reacts. The entire test took 559.4seconds (9minutes and 19.4seconds).

It can be seen that  $D_S$  struggles with tracking the set point at low values and have relatively large overshoot, perturbations and stationary deviation. Apart from the delay initially the error is at a maximum of approximately 11%. This is discussed further in chapter 5.1.

This following graph (4.14) shows the rapid and violent perturbations of the estimate ( $D_{S,est}$ ). The perturbations becomes bigger at higher values for the set point in contrary with  $D_S$ . These values are used as feedback for the PID controller in this test.

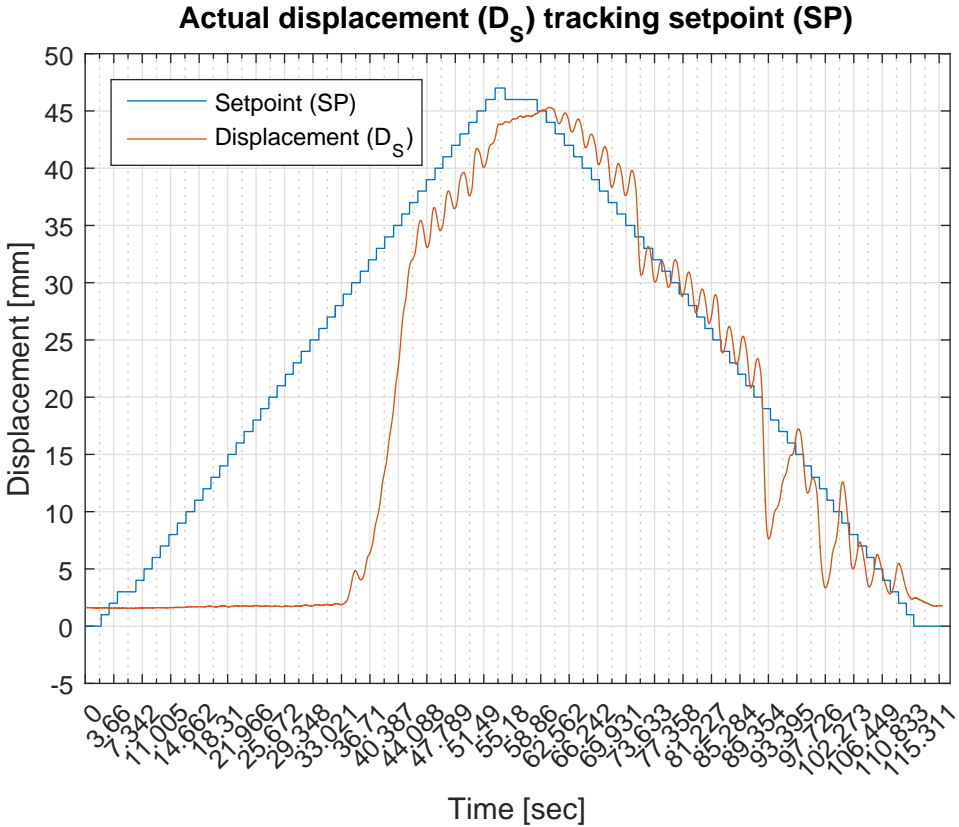


**Figure 4.14:** PID-test1:  $D_{S,est}$  tracking of set point with  $D_{S,est}$  as feedback.

The performance of the estimate is poor with rapid and violent perturbation, but the plot is cropped to the same range in  $x$  and  $y$ -axis for comparison with figure 4.13.

**PID-test2: *Small steps up and down***

$K_p = 4$ ,  $T_i = 40$  and  $T_d = 1$ . This graph shows the performance of the actual displacement ( $D_S$ ) responding to smaller changes in set point. Compared to the plots shown in figure 4.13 these are better. However, they are still not good.

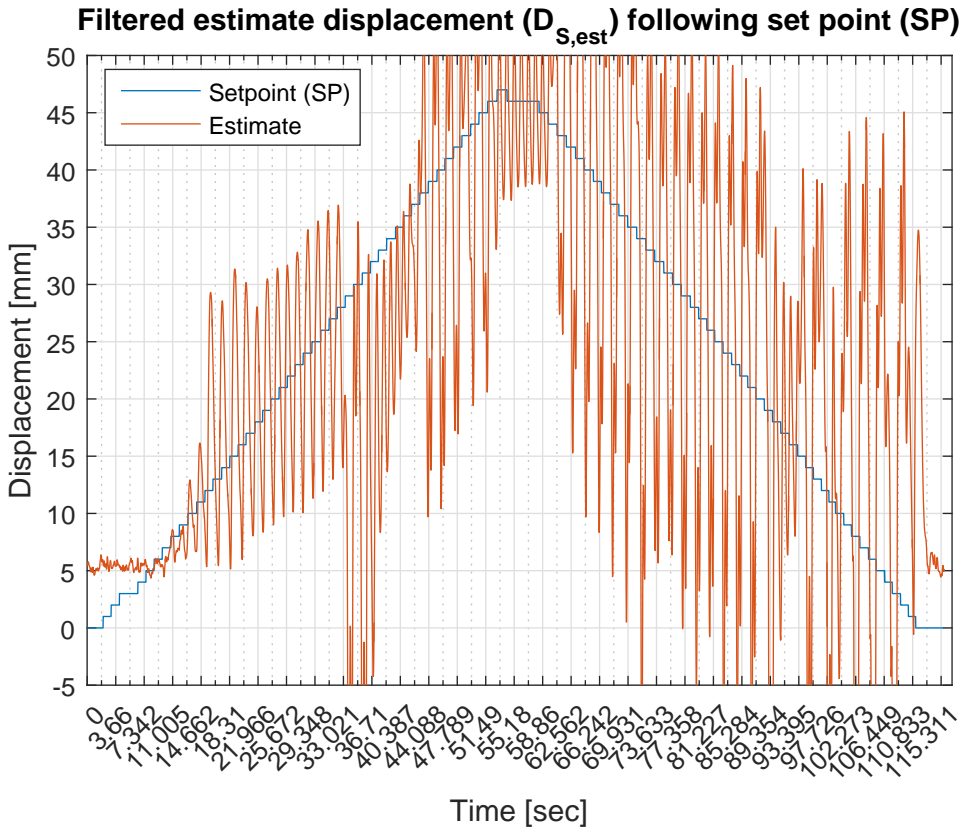


**Figure 4.15:** PID-test2:  $D_S$  tracking of set point with  $D_{S,est}$  as feedback.

It took 33.3 seconds before  $D_S$  reacts and the entire test took 115.3seconds (1minute and 55.3seconds).

The performance bears resemblance to the former plots with regards to the delay in response of  $D_S$ . Also, the response oscillates with an amplitude of approximately 1mm and has an offset from the set point with around 2 - 3mm while contracting. The tracking is better when the displacement is decreasing. The reason for the delay and oscillation will be discussed in chapter 5.1.

This graph shows the perturbations tracking the set point poorly, but again still slightly better compared to the plots shown in figure 4.14. These values are used as a feedback to the PID. The performance will be discussed in chapter 5.1.



**Figure 4.16:** PID-test2:  $D_{S,est}$  tracking of set point with  $D_{S,est}$  as feedback.

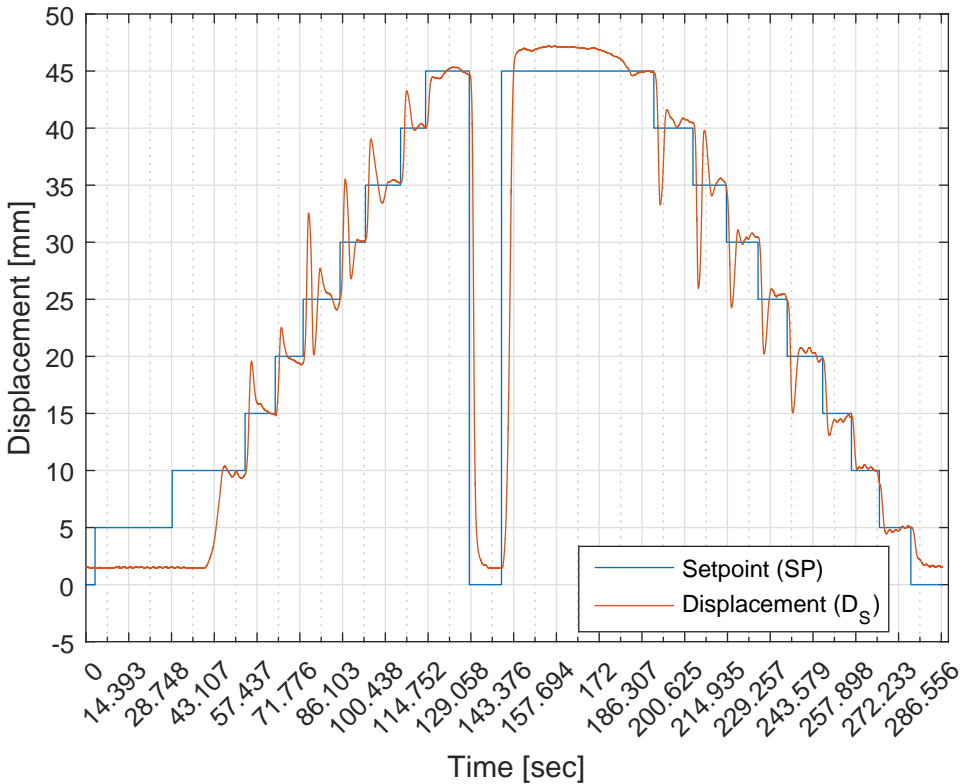
The plot is cropped to the same range in the  $x$  and  $y$ -axis as in figure 4.15, even though the estimate exceeds the range, to compare the two plots more easily.

### 4.2.2 Actual $D_S$ feedback

In the following tests the actual displacement ( $D_S$ ) is used as the feedback. Thus, the estimate is redundant and not used for anything other than to show whether or not the estimate improves when the feedback is smoother.

**PID-test3: *Medium steps up, step down, then step up and medium steps down (Actual  $D_S$  feedback)***

$K_p = 1.6$ ,  $T_i = 120$  and  $T_d = 0.05$ . This graph shows the tracking capability of  $D_S$  when the input sequence in set point is the same as in figure 4.13.



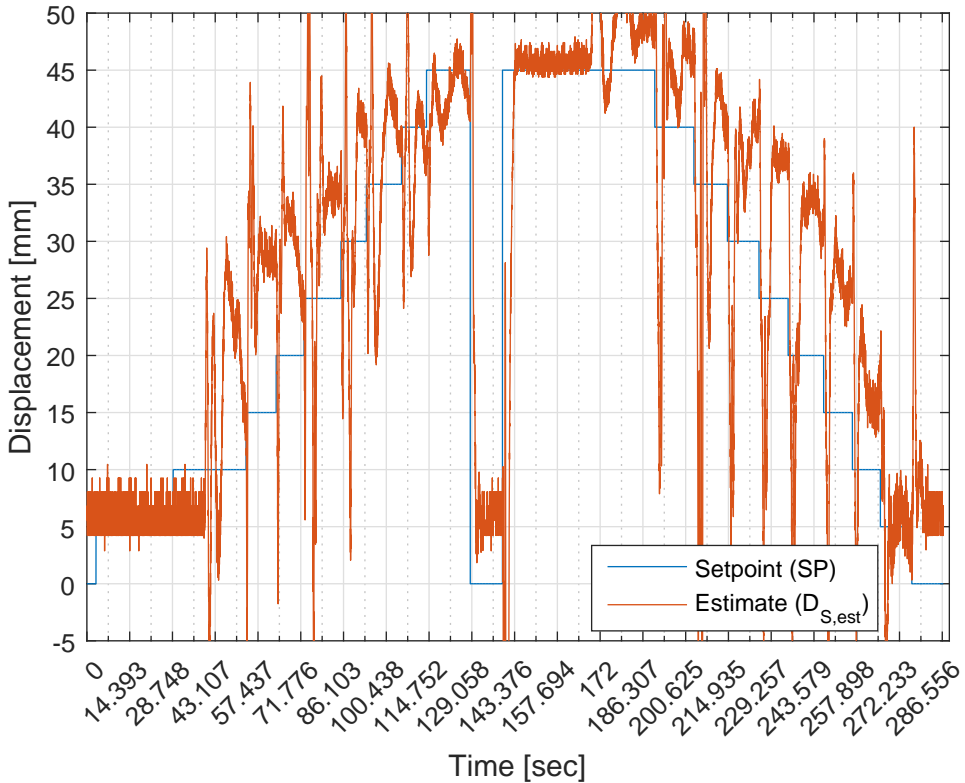
**Figure 4.17:** PID-test3:  $D_S$  tracking of set point with actual  $D_S$  as feedback.

$D_S$  took 40.4seconds to react, and the test took in total 286.6seconds (4minutes 46.6seconds). The performance will be discussed in chapter 5.1.



The performance of this test is significantly better than the plots in chapter 4.2.1. There is approximately 18% of the set point range overshoot of the 5mm steps at worst, but the settling time is relatively low, with approximately 10seconds at worst.

This graph (4.18) shows that the estimate performs poorly also now. The overshoot is very large, the settling time is difficult to calculate as the perturbations are too rapid and violently, and  $D_{S,est}$  has an offset from the set point. The performance will be discussed in chapter 5.1.

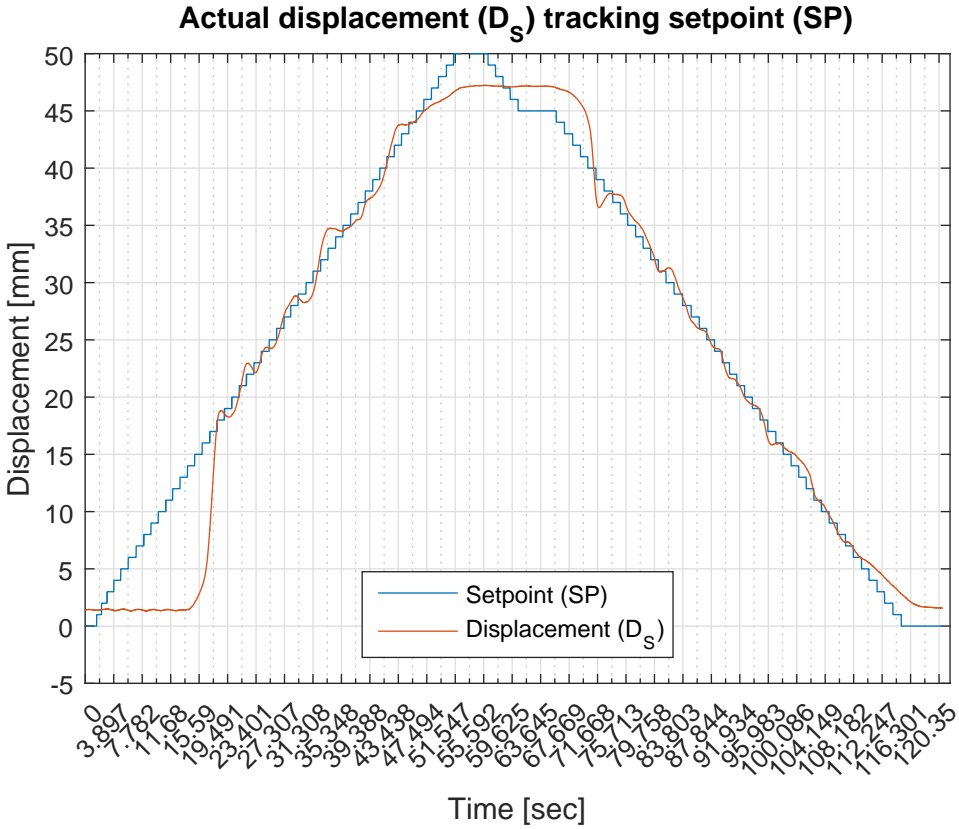


**Figure 4.18:** PID-test3:  $D_{S,est}$  tracking of set point with actual  $D_S$  as feedback.

These values of  $D_{S,est}$  are not used for anything other than for comparison with the results done in chapter 4.2.1.

**PID-test4: *Small steps up and down (Actual  $D_S$  feedback)***

$K_p = 4$ ,  $T_i = 10$  and  $T_d = 1$ . This plot shows that the actual displacement ( $D_S$ ) tracks the set point relatively well, apart from the late reaction in the start.

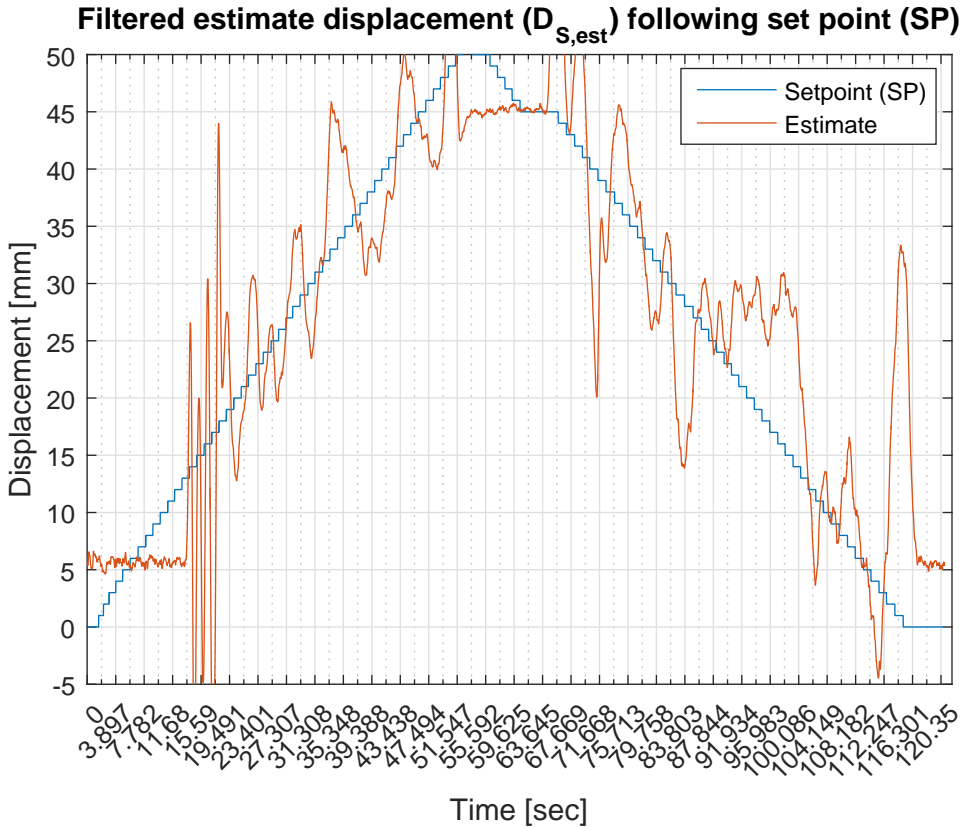


**Figure 4.19:** PID-test4:  $D_S$  tracking of set point with actual  $D_S$  as feedback.

It took 14.5 seconds for  $D_S$  to react and the test took 120.4seconds (2minutes and 0.4seconds) to conduct.

There is a bit of overshoot when the set point decreases again. The performance will be discussed in chapter 5.1.

This graph shows that the estimate ( $D_{S,est}$ ) still struggles with tracking the set point. There are large and sporadic perturbations. However, these values are not used for anything. They are just used to compare with the performance of the estimates in chapter 4.2.1.



**Figure 4.20:** PID-test4:  $D_{S,est}$  tracking of set point with actual  $D_S$  as feedback.



# Chapter 5

## Discussion

In this chapter the different challenges encountered through the project period are brought up. The discussions will revolve around why it was considered a challenge, potential different approaches and how this affected the overall solution compared to different proposed solutions. Lastly, future applications and the potential of the SMA actuator is discussed. The arguments will be based both on theory and subjective thoughts and opinions the author of this thesis has formed through the work with this project.

Late in the project period other methods for modeling a hysteresis function was found. An article by Hughes and Wen (1997) suggests the Preisach model for the hysteresis in piezoceramic and SMA materials. This model has not been researched, but could potentially have simplified the implementation of a mathematical model.

The discussion is arranged as follows: Firstly, the performance of the PID-tests are assessed, and possible reasons for the behavior are discussed. Secondly, the spikes in the estimate of the calculated resistance ( $R_W$ ) of the SMA actuator. Then, the mathematical model is discussed, where consequences for not implementing it are considered and alternate implementation methods are mentioned. Next, the issue of the measurement timing of the PWM signal and the *analog read* with the Arduino. Following this is an evaluation of the solution with regards to both the LabVIEW code, the electrical circuit and the total solution seen as a whole. Lastly, the application potential is discussed, where possible future implementations are mentioned as well as implementations in development.

### 5.1 PID test performance

Implemented in this system is a simple PID controller, just like the ones used in Lee et al. (2013), Dutta and Ghorbel (2005), Liu et al. (2009) and Senthilkumar et al. (2011). The idea was to implement a simple controller with the capability of bringing the displacement ( $D_S$ ) to its set point based on the estimate. This means that for a signal which is as noisy as the estimate calculated in this thesis ( $D_{S,est}$ ), the controller must be fast enough to counteract these perturbations. The controller could beneficially have been a PI (Proportional Integral) controller, such that the responsiveness of the system would be as fast as possible. However, setting the derivative term ( $D$ ) to zero would increase the settling time of the system.

In this section the performance of the controller tested throughout chapter 4.2 with regards to accuracy, noise and responsiveness, and will be judged based on the authors subjective evaluation.

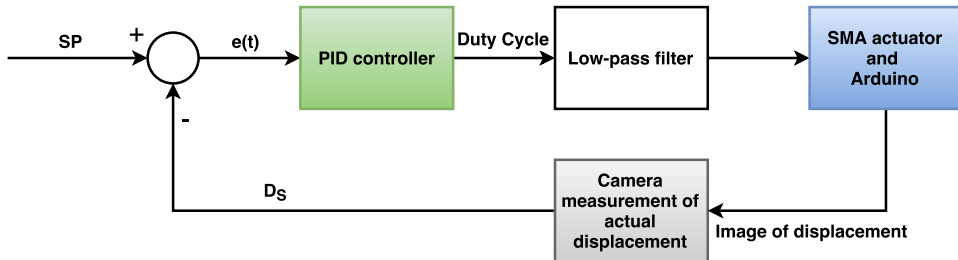
The first two tests are performed with the  $D_{S,est}$  as feedback and a particular input sequence for the set point, then the same two tests with the same input sequence are conducted with  $D_S$  as feedback. This is done to compare the performance of the system if for instance the feedback was as accurate as the actual displacement ( $D_S$ ). Ideally the PID parameters should have been tuned to a more optimal performance, but since the controller was completed late in the project period this was not the focus. The PID parameters have been tuned by eye based on the evaluation of the author of this thesis. Also, the input sequence should have been inputted as a preprogrammed sequence, rather than changing the input while the program is running by hand, such that the test lasts equally long and the distinction in responsiveness becomes more apparent.

Firstly in chapter 4.2.1, a test was conducted where both a stairs and a step function was inputted. It was observed that the system would not react before the set point would reach values of around  $30mm$ . One possible reason for this is that the large perturbations of the estimate are mainly below the set point at these values, such that the accumulated error causes the integrator of the PID to have a build-up in the wrong direction. Thus, it takes a longer time to build up the needed gain to compensate for the increasing error in the opposite direction. This is, however, out of the two different tests that were conducted on the actuator, the worst performing estimate. Therefore, the performance is to a certain degree relatively satisfying, considering that  $D_S$  is tracking the set point to some extent. The tracking ability can be seen in figure 4.13. Had the estimate been filtered such as shown in figure 5.5, the performance most likely would have improved significantly, but this was not done until after the tests were conducted. The total time of the test was  $559.4seconds = 9min$  and  $19.4seconds$ , which is a relatively long time, but this would improve if the effect input to the wire increased closer to the maximum values proposed by Dynalloy (2016).

Compared with the tracking ability of the tests conducted in chapter 4.2.2, where  $D_S$  was used as the feedback, the aforementioned performance is quite poor. However, the system in “*PIDtest3*” still struggles with reaching the set point at values in the range of  $5 - 10mm$  when contracting. This is partially because the integrator gain ( $K_i$ ) is set too low such that it takes time to build up the input to have a fast response to lower values. Also, the wire must be heated up to a temperature high enough to cause a contraction regardless of the set point. This means that the wire will reach this temperature faster if the set point is further away from the initial, fully elongated length of the actuator because the momentary error is high causing

the proportional gain ( $K_p$ ) to quickly set a high duty cycle of the PWM signal.

The estimate becomes smoother because  $D_S$  is used as feedback rather than  $D_{S,est}$ , if figure 4.18 and 4.20 are compared with figure 4.14 and 4.16. This is because a smoother signal is sent to the Arduino, meaning the duty cycle does not change as rapidly and violently, and consequently the calculated parameters does not vary as much either. Even so, the estimate has a relatively large stationary deviation from the set point compared to other tests. However, this will not affect the system in any way since  $D_{S,est}$  is not used in the feedback loop, as illustrated in figure 5.1. The total time of the test was  $286.6seconds = 4 min$  and  $46.6seconds$ , which is about twice as fast as the system with  $D_{S,est}$  as the feedback. Though this is dependent on how the person conducting the test considers the settling time, it serves as an indication that the responsiveness improves when the feedback is smoother and more accurate.



**Figure 5.1:** Block diagram of the feedback loop with  $D_S$  as feedback. Designed in *draw.io* (JGraph, 2016).

Further, a similar test were conducted, but the set point now changed with smaller steps from  $0mm$  to the maximum value of about  $46mm$  with  $1mm$  steps, and the same small steps down to a fully elongated state again. The large steps was removed and the intervals between the small steps were shortened down to approximately one second regardless of the settling time of the response. In this way the ability to track rapid changes would be tested. In this test the struggle of tracking the lower values at an early stage of the test became even more apparent.  $D_S$  would not react until the set point reached values close to  $35mm$ , and would not completely reach the set point even then. The main reason for this is that the initial value of the estimate has a higher value than the set point ( $SP$ ) which it holds until  $SP$  passes  $D_{S,est}$  at around  $5mm$ . This is, as mentioned earlier, causing the integrate part of the PID to build up a negative gain which take longer time to cancel.

When the feedback is switched to  $D_S$  the tracking performance improves significantly. The fact that the system does not react before the set point reaches values over  $15mm$

is still an issue, because of the same reasons as mentioned earlier. However, in the case where the actual displacement ( $D_S$ ) is used for the feedback the zero value for when the actuator is fully elongated can be offset. This can be calibrated in the LabVIEW program, but was not noticed until late in the project period. Reasons for this offset can be physical contact with the camera or a change in ambient temperature.

The small perturbations in the tracking, seen in figure 4.15 and partially in figure 4.19, are caused by a phenomenon called “*derivative kick*”. This is when the small step in set point causes the error input of the PID ( $e(t)$ ) to momentarily increase, thus making the derivative of the error spike (or “kick”) in the opposite direction to dampen these rapid changes. Mathematically, the steps in the set point are not continuously differentiable. This phenomenon can be solved by inputting just the negated process variable ( $-D_{S,est}$ ) instead of the error ( $e(t) = SP - D_{S,est}$ ), thus nullifying the issue with value changes that causes a high derivative. Alternatively, the derivative term ( $K_d$ ) can be set to zero, effectively canceling out the  $D$  term completely, and using a PI controller instead.

Also, a second order low pass filter reference model could have been used for creating a smooth reference set point for tracking control. In this way the steps in  $e(t)$ , as a result of changing the set point with a step, are smoothed in accordance with the reference model. The simplest form of a such a model is using a low-pass filter structure. A commonly used method to generate this smooth reference trajectory for the set point is, according to Fossen (2011), to use a model motivated by a physical system, like for instance a *mass-spring-damper* system, which has the form of a second order low-pass filter.

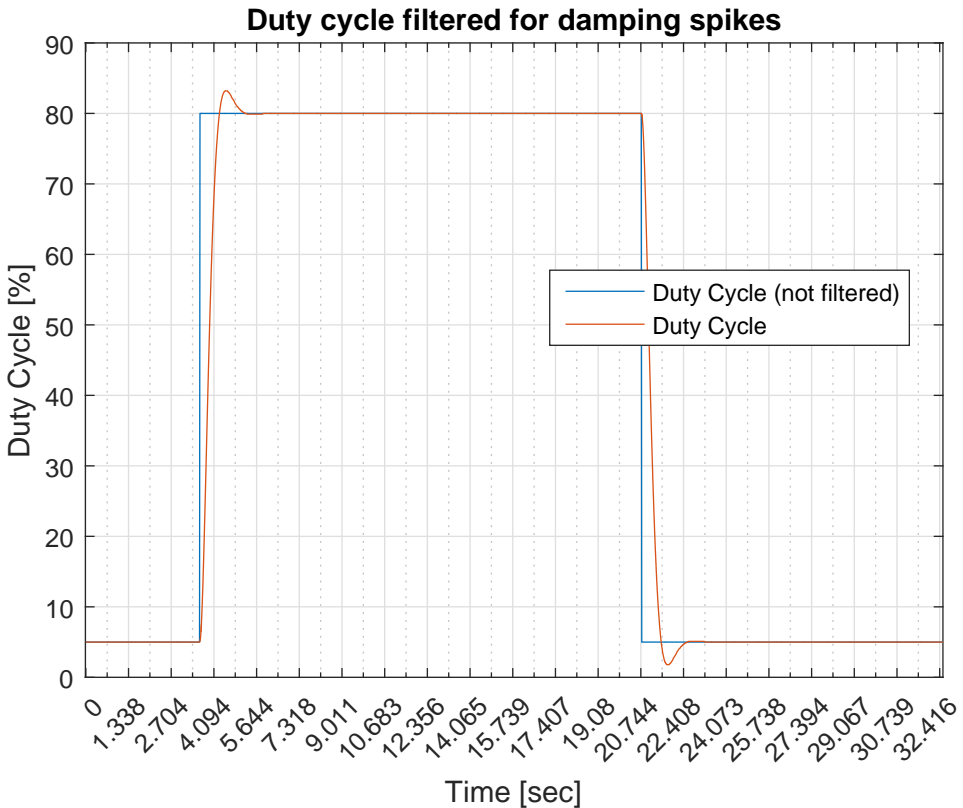
Alternatively, a feedforward controller can be used to track the set point. However, this is dependent on having a model of the system since the feedforward model often is the inverse of the plant itself. Ideally, this makes the system track the set point with no error. According to Mark W. Spong (2005) it is a commonly used for tracking of time varying set points. This is the approach used in Liu et al. (2009).



## 5.2 Spikes in estimate

In this section the spike in the estimate and filtering the duty cycle will be discussed. As mentioned in chapter 3.1 and 4.1.4, there was observed a spike in the calculated resistance value ( $R_W$ ) when the duty cycle changed rapidly, which caused an even bigger spike in the estimated displacement ( $D_{S,est}$ ). However, this spike in resistance value is not real. The actual behavior of the resistance ( $R_W$ ) would start of at around  $55\Omega$  and decrease to approximately the same value as shown in figure 4.7 or 4.9, which is approximately  $49\Omega$  and  $50\Omega$ , respectively.

The spike occurs because the response time of the voltage drop ( $V_{R_W}$ ) compared to the current ( $I_C$ ) are different. Since the calculated resistance in the wire expresses the relation between these parameters, ( $R_W = \frac{V_{R_W}}{I_C}$ ), will have a large value when  $V_{R_W}$  has a high and  $I_C$  has a low value.



**Figure 5.2:** The step in duty cycle inputted to the estimate filtered.

The issue was temporarily solved by filtering the duty cycle such that the two values responded more evenly. As seen in figure 5.2 the filtered input have some overshoot. Alternatively, a reference model structured as a second order low-pass filter could have neglected this spike because a first order LP-filter is not continuously differentiable.

### 5.3 The mathematical model of hysteresis

Firstly in this section, the difficulties with implementing a mathematical model of the system based on the Duhem model for hysteresis and the consequences of not implementing it will be discussed.

This became a problem mainly due to the complexity of the model. A lot of time was spent trying to duplicate the work done in Liu et al. (2009) and Dutta and Ghorbel (2005), such that the model could be verified before it was implemented with the rest of the system. A *SIMULINK* model was tested based of the suggested equations in the articles, but gave inconclusive results. Therefore, the *SIMULINK* model will not be included in this thesis.

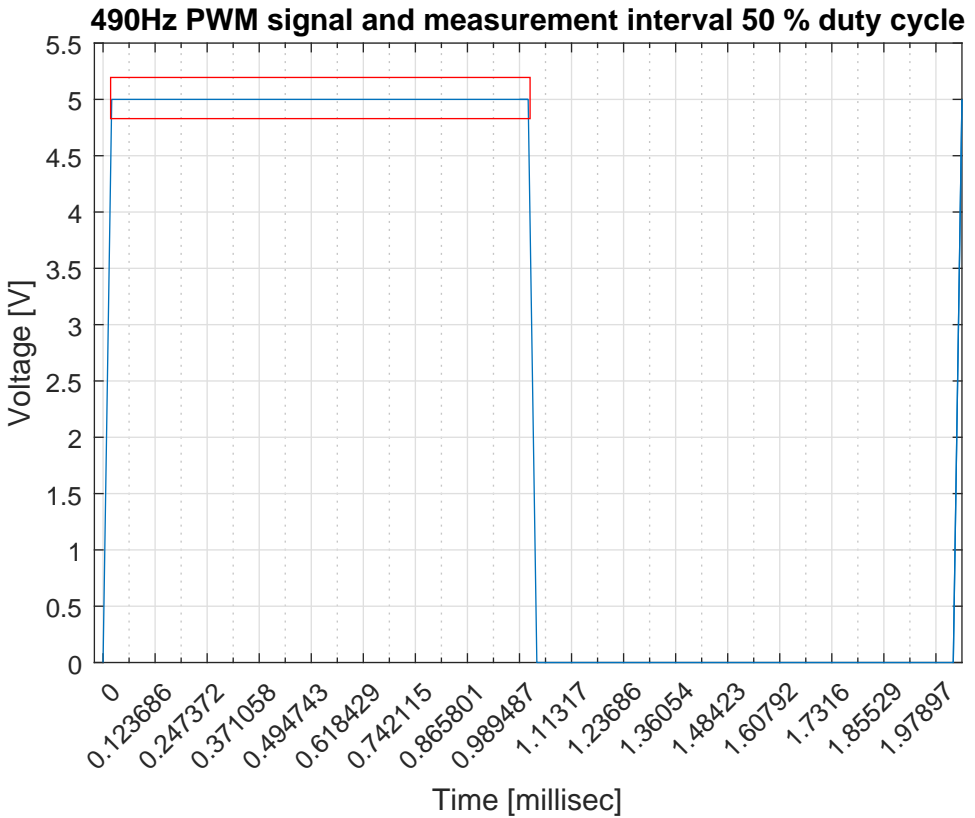
The main problem occurred when the minor loop (see chapter 2.2.6) was supposed to be included in the model. Model parameters were difficult to relate to the physical meaning as the model became more complex, and thus, tuning them to get the wanted behavior proved problematic. Alternatively, the Preisach model from the article by Hughes and Wen (1997), could have been researched and potentially implemented in stead of the Duhem model.

The consequences of not implementing a mathematical model for the system is mainly the drop in accuracy and responsiveness. The inverse compensator introduced in Liu et al. (2009) makes the system able to affect the input to the SMA actuator by compensating for the hysteresis and the dynamics of the wire, as well as the bias force, the dimensions of the wire, thermodynamics and electrical power required to produce movement. This allows for a fast and accurate tracing of a input sequence for the set point. However, the high accuracy is not as important in the case where a tactile sensor is implemented. When the tactile sensor also is a part of the feedback, the SMA actuator will tighten or contract until the desired set point in grip force is reached, regardless of the angle of the finger in the end effector. Therefore, the system does not require accuracy down to a millimeter. The accuracy is more important if the actuator is implemented in for instance a locomotive mechanism for endoscope as done in Kim et al. (2005).

Also, the possibility of simulating the performance of the SMA actuator and then testing different controllers to validate and possibly improve the overall performance is a desirable benefit when implementing an accurate model of the system.

## 5.4 Measurement timing

The root to a lot of the problems regarding faulty measurements was timing the analog read with the PWM signal. The reason for this being that when the PWM signal is low the circuit is essentially off, and the measurements are biased by the power supply charging up the capacitor in the low-pass filter in the circuit.



**Figure 5.3:** Plot of the timing interval of measuring the analog read. Red rectangle marks measurement area.

Figure 5.3 shows that the program has a  $1\text{ms}$  measurement interval at 50% duty cycle. Which means that for duty cycles down to 8% yields a measurement interval of  $2\text{ms} \cdot 0.08 = 0.16\text{ms} = 160\mu\text{s}$ . When running the version of the LabVIEW code with the *soft PWM* this was a part of the issue. Then the program could only produce a PWM base frequency of  $100\text{Hz}$  which proved to be low, as the performance of the SMA actuator was jerky, meaning that it was not smooth. The base frequency could possibly have been lower than  $490\text{Hz}$ , but the reduced functionality with the LINX

package in LabVIEW would not allow changing the default setting for PWM base frequency.

This means that, there is a trade-off with respect to producing a high enough PWM frequency, but still having it low enough to ensure a measurement of the PWM signal when it goes high.

## 5.5 LabVIEW code

First of, the LabVIEW code will be assessed isolated considering what worked well, what did not work well and what could have been done differently.

The part of the solution regarding the LabVIEW code that work well is the fact that the displacement actually tracks the set point to a certain degree even though the estimate is poor. Additionally, the actual measurement of the displacement with the camera works very well. As can be seen in plots of the actual tracking for  $D_S$  as well as the displacement in the contraction phase in figure 4.4, there are little to no measurement noise in these plots. The LINX package for LabVIEW made the communication between the code and the Arduino program simple and effective, and neglected the time lag between the two.

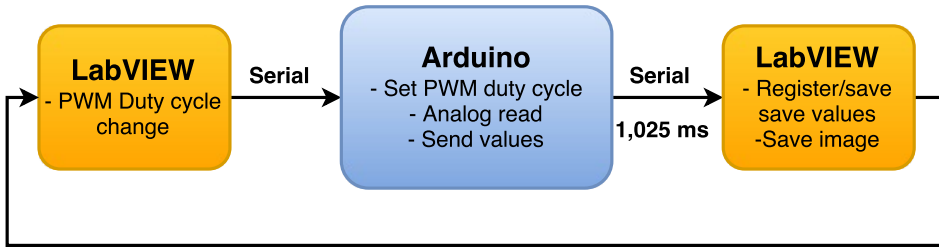
Contrary to the aspects that worked well are the estimate of  $V_{tot}$  and the amplification compensator. Even though both of these functions serves their purpose relatively well, combined they make the system very sensitive to rapid changes in duty cycle. Consequentially, this also affected the estimate of the displacement ( $D_{S,est}$ ) which also performed poorly.

In hindsight, the estimates for the compensator could have been constructed differently. For instance, it could have been a linear function which converts the *raw data* measured with the Arduino into the real value similar to the one measured with the multimeter, regardless of the duty cycle. The equation is expressed as follows:

$$y = 0.37x - 0.46 \tag{5.1}$$

Where  $y$  is the real value of the voltage drop ( $V_{R_k}$ ) and  $x$  is the *raw data* of the analog read from the Arduino. Additionally, the estimate ( $D_{S,est}$ ) should have been filtered with a lower cutoff frequency, such that the rapid and violent perturbations gets attenuated and thus make the estimate smoother.

Figure 5.4 illustrates where the delay happens. Figure 3.4 shows the time delay between setting the duty cycle in the LabVIEW code and then register the same change in duty cycle in the LabVIEW code after it was sent to the Arduino.



**Figure 5.4:** The flow of the sending and receiving of the values. The orange blocks represent the LabVIEW code and the blue block represent the Arduino code.

By implementing the LabVIEW package LINX, the delay is negligible. One drawback of using LabVIEW and especially LINX is the drop in functionality, meaning such options as changing the PWM base frequency and other registers to customize the PWM signal, thus being able to implement a soft PWM which can produce a fast enough base frequency. These are functions that would have been useful. Hence, LabVIEW might not be the ideal program for this task. If the solution is intended to be small and compact it would be beneficial to be able to upload the code and have the program running without having to be linked to a computer. Therefore, an Arduino or a different micro controller could have benefited the practicality of the solution by for instance making a soft PWM in the Arduino or *C*-code. The micro controller would still have to communicate with a device which is supplying the set points, but this could be transferred through any desired communication protocol.

Nonetheless, LabVIEW serves as a sufficiently good development platform, and to neglect the time delay, LINX serves as a quick and efficient alternative to more advanced ways to insure fast communication with an easy implementation.

### 5.5.1 Displacement estimate

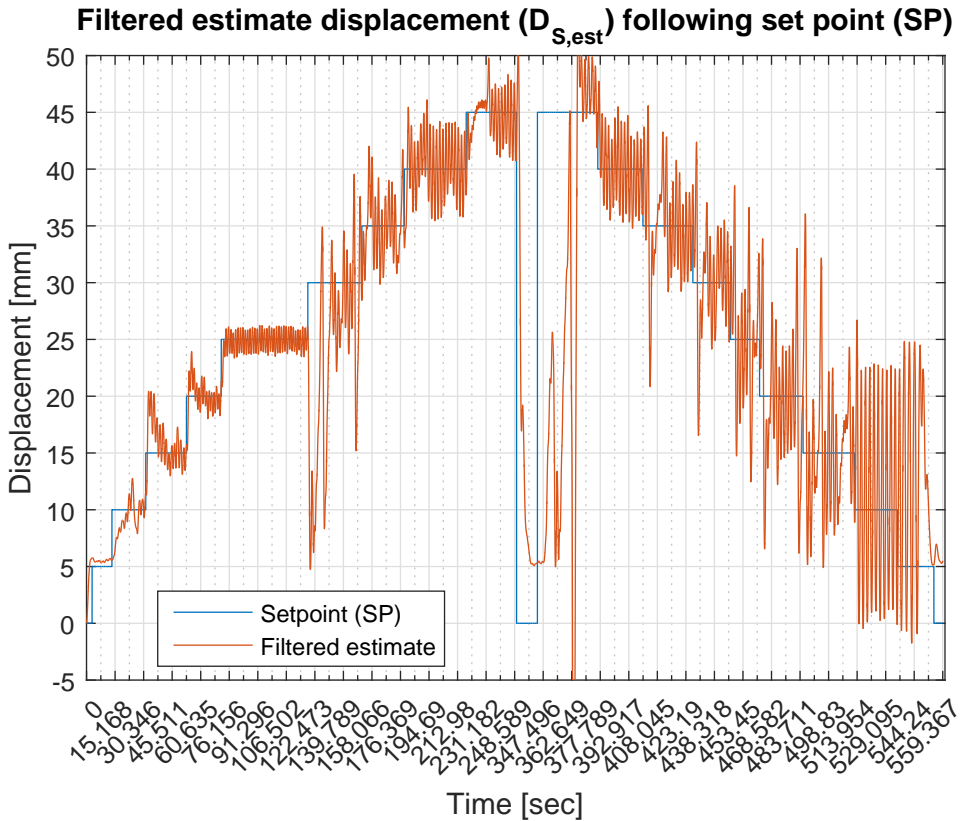
The estimated displacement ( $D_{S,est}$ ) is based on the near-linear part of the hysteresis characteristics in the contraction or heating phase. This is because the actual characteristics of the relation between the displacement ( $D_S$ ) and the resistance ( $R_W$ ) has a hysteretic curve seen in figure 4.2. The perturbations at the start of both the contraction and elongation phase is due to the current starting at low values and perturbing as the length of the wire changes. Additionally, the polynomial fitted function can be of a lower order.

The performance of the estimate ( $D_{S,est}$ ) however, is highly sensitive to variations in duty cycle values because of the variations in the estimate of  $V_{tot}$  and the amplification compensator. The displacement estimate is dependent on both of these estimates,

which are amplified throughout the circuit. The displacement estimate is illustrated in figure 3.2.

Lastly,  $D_{S,est}$  should have been filtered to attenuate the rapid and violent perturbations of the estimate. Figure 5.5 shows the estimate from in figure 4.14 filtered with a lower cut-off frequency. Considering the tracking capabilities of the controller in chapter 4.2.1 it is reasonable to think that the tracking capabilities would improve significantly with regards to settling time and also stationary deviation.

However, since the controller was implemented successfully at a late stage in the project period the strict filtered estimate was not implemented.



**Figure 5.5:** PID-test3:  $D_S$  tracking of set point with  $D_{S,est}$  as feedback and the estimate is in this case filtered.

## 5.6 The circuit

Firstly, the circuit will be assessed isolated with regards to what worked well, what did not work as well and what could have been done differently. Then, the strengths of the circuit is that it is simple in principle and consists of affordable components. Also, it actually performs its task, to a certain degree, which is to measure the voltage drop over the resistor,  $V_{R_k}$ .

The biggest weakness of the electrical circuit is the low-pass filter which introduces a bias in the amplification from the capacitor when the duty cycle is low. The reason for this is discussed in chapter 5.6.1 below. Additionally, the circuit should have been designed to drive the current through the wire closer to the recommended maximum value such that the responsiveness of the system would improve. This means that either the supply voltage is set higher and/or the known resistor  $R_k$  is set lower. However, the resistor is already relatively low, which means that it would be more efficient to increase the supply voltage.

Since the issue with the amplification occurs when the capacitor is charged up by the supply voltage through the feedback resistor of the differential amplifier when the PWM signal is low, an alternate circuit could have been designed. Ideally, the filter should not have to be implemented, but if the problem should be fixed only by changing the hardware, the capacitor should be bypassed when the PWM signal get low. This could have been done by switching out the filter with a different transistor at that period. However, this approach have not been researched, but is mentioned as an alternative.

Having a differential amplifier cascaded with an active low pass filter with amplification may not be the optimal solution for measuring and filtering a voltage drop over a resistor. When the signal gets amplified two times and filtered, it can be difficult to distinguish between the different amplifications. In this case the differential amplifier amplifies the voltage drop and then after filtering the PWM the signal gets amplified again. A possible alternative to this could be to have a *Unity Gain Differential Amplifier* such that  $V_{R_k}$  gets referenced to ground and then filtered and amplified. This means that all the resistors have the same ohmic value, and the differential amplifier consequently has no gain ( $A_{diff} = 1$ ). The difference between the two inputs (see equation 2.6) is then fed trough. In this way the total amplification is easier to control and the filter does not need to smooth an amplified signal. Additionally, the gain of the active filter will make sure the signal range increases and thus, the corresponding resolution of the 10-bit ADC in the Arduino.

It would also be beneficial to include an extra differential amplifier to measure  $V_{tot}$  instead of estimating it. It would be less calculations in the LabVIEW program and the measurements would have been more reliable. However, this was suggested late

in the project period and at that point all the operational amplifiers were taken and there was no time to get a new one. Therefore, the solution is called *temporary*. The reason for removing it was that before implementing the filter, the voltage drop over the Darlington transistor ( $V_{CE}$ ) was constant ( $\approx 0.75V$ ), and thus  $V_{tot}$  was also constant ( $\approx 19.25V$ ). This meant that the differential amplifier was redundant, and was consequently removed.

Beneficially, the input effect of the wire should have been closer to the recommended limitations mentioned in Dynalloy (2016), such that the responsiveness of the wire would increase. This could have been done by allowing the duty cycle to go closer to its maximum of 100%. However, because of the low-pass filtering of the duty cycle in the LabVIEW code, the duty cycle input have some overshoot. This could cause problems if the input would give a step from a low duty cycle to 100% and the filter had an overshoot causing the PWM register value to overflow and start from the bottom value of 0%. The higher the duty cycle, less overshoot is allowed. Therefore, rather than maxing out the duty cycle to get a higher effect input to the SMA actuator, the supply voltage could increase or the value of the known resistor could be decreased. In this case the amplification would also have to be adjusted, such that the signal lies in the 0 - 5 V range of the Arduino ADC.

### 5.6.1 Low pass filtering and amplification error

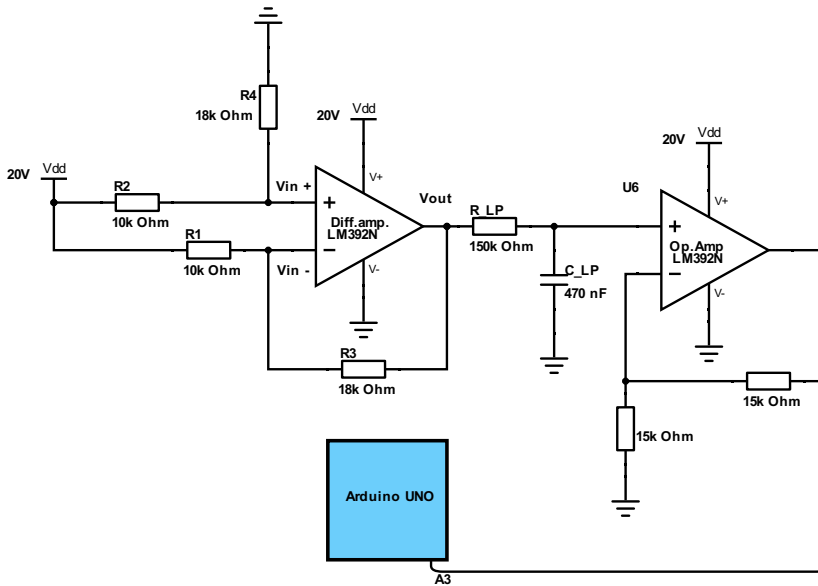
Initially meant as a temporary solution to the timing issue of the PWM signal, a first order active low-pass filter with amplification was implemented in the circuit. A potential problem with this could have been the phase shift it introduces to the filtered signal, in accordance to equation 2.11. However, since the signal is not dependent on the phase of the signal and the fact that the filtered signal was smoothed to approximately a DC signal, this would not prove to be an issue. Nevertheless, it was important to set the cut-off frequency low enough to flatten the PWM signal to a signal without too much ripple. These ripples could potentially cause measurement error. However, the filtering will increase the complexity of the circuit with respect to the Darlington transistor and especially the amplification.

The main problem occurs when the PWM signal goes low. Figure 5.6 is the equivalent circuit of the final experimental electrical circuit when the PWM signal goes low. This means that the Darlington transistor turns off, and subsequently there are no current through the wire ( $R_W$ ).

Because of the non-inverting input of the differential amplifier is configured as a voltage divider to ground, the drop from the supply voltage over  $R_2$ , causes  $V_{in+}$  to have a  $20V \frac{10k\Omega}{10k\Omega+18k\Omega} = 7.14V$  input. The operational amplifier will then force the  $V_{in-}$  to have the same value. This means that the 20 V supply voltage has a voltage drop of  $20V - 7.14V = 12.86V$  over  $R_1$ . Where the voltage drop  $V_{R_k}$  in this case



is considered negligible. From this, the current through  $R_1$  and  $R_3$  is calculated to  $1.3mA$ . This means that the voltage drop over  $R_3$ , called  $V_{out} = 18000\Omega \cdot 0.0013A = 23.4V$ , which is higher than the saturation voltage of the differential amplifier. Regardless, this voltage charges up the capacitor ( $C_{LP}$ ), which is consequently supplying the signal when the duty cycle is low offsetting the voltage read by the operational amplifier.



**Figure 5.6:** Equivalent schematic of the circuit when PWM signal is low. Designed in *DigiKey* Digikey (2016).

Seeing as the a low-pass filter outputs the average voltage of the PWM signal as a DC (Direct Current), the calculations done in table 4.2 are correct in that way. The multimeter also smooths out the PWM signal, making the values comparable. Shown in figure 4.1 the amplification converges towards the constant value of  $D_{diff} = 1.8 \cdot 2 = 3.6$

A soft PWM solution with a base frequency high enough that when inputted in the SMA actuator, the changes in displacement are smooth, would make the low-pass filtering redundant and is considered by the author of this thesis as the optimal solution to this problem. The measurements would in that case always get timed

when the PWM signal is high, thus making the LP-filter unnecessary.

The consequences of biasing the amplification is that the calculated value of  $V_{R_k}$  is wrong not the duty cycle is at 100%.

## 5.7 The total solution

The term “*total solution*” covers all of the aspects of the solution as a whole, which comprises the electrical circuit and the LabVIEW program.

In this section the total solution will be assessed with regards to what worked well, what did not work as well and what could have been done differently.

The system seen as a whole, considering the simple design of the different aspects of the solution, performs relatively well. The set point is tracked with a low accuracy, but when using a tactile sensor the requirements with regards to accuracy is not as strict. Additionally, the GUI designed in LabVIEW is intuitive and makes the program easy to use.

On the other hand, the estimate is not good enough mainly, because it is dependent on many other inaccurate estimates such that the system becomes highly sensitive. This system would not be applicable to tasks that requires high accuracy.

Since the low-pass filter is the root of most of the problems it should have been excluded from the solution. Instead, a program based on a soft PWM design should be implemented. In this way the analog read will always happen when the PWM signal is high. Then the low pass filter is not needed, and the system is significantly simplified because there is no capacitor to bias the amplification. If designed properly LabVIEW can be used to run the soft PWM code, but it can prove to be problematic if the LINX package is used. Therefore, the communication between LabVIEW and the Arduino must be optimized to minimize the loss of data and time delay. Alternatively, an Arduino or an equivalent micro controller should be implemented such that a soft PWM could be implemented. Also, a GUI should be designed for the user to easily tune the controller and input a set point.

## 5.8 Application potential

The SMA actuator has great potential and is highly usable for slow to medium frequency tasks which are not to temperature sensitive. Because of its low *power-to-weight ratio* and the sensor-less feedback, the possibilities are many. Research have been done in designing prosthetic hands which utilizes SMA actuators to simulate muscles in the fingers of a five fingered hand (Laurentis and Mavroidis, 2002), or a brace for active gait assistance for people with neuromuscular disorders, such as described in Stirling et al. (2011). Other research revolves around fiber optics, such as the article by Kim et al. (2005) where an endoscope was developed with SMA actuators as a part of the solution to make maneuvering easier. Endoscopes are medical devices used to diagnose diseases in the gastrointestinal tracks. This type of implementation can be used in all kinds of fiber optics or maneuvering of long thin objects.

Because of the light weight og the SMA actuator it could possibly be implemented when designing robotic insects. If the control system for the displacement was perfected it could actuate the legs of the insect. If the responsiveness got improved and the wire could contract and elongate at high speeds, it could also actuate the wings. Alternatively, Kim et al. (2006) designed an earthworm-like micro robot where the SMA actuator made the worm contract and extend to emulate the movement used by worms. Other insect designs include a simple inchworm design and a Phasmatodea, commonly known as a walking stick.

In an article by Cianchetti et al. (2014), they have developed a mechanical *soft robotics* arm inspired by an octopus tentacle. The term “*soft robotics*” describes robotics that are composed of very light, flexible and soft material. By shaping the SMA actuators as springs and arranging them in a particular way the arm is able to perform tasks such as grasping and multi-bending gestures, under water.

The wire could potentially replace any mechanically actuated process if the behavior of was optimized, meaning characteristics such as the range of motion and the responsiveness was increased. Additionally, the elongation phase of the wire must be optimized, either by using different wire as an antagonist to actively stretch out the wire or by implementing a active cooling system to cool down the wire in the elongation phase. To increase the force produced by the contraction of the wire research done by Mosley and Mavroidis (2001) shows that if several wires are woven together in a counter rotating helix pattern around supporting disks, it will improve the strain capabilities.



# Chapter 6

## Conclusion

This thesis investigated the possibility of designing a sensor-less displacement control of a Shape Memory Alloy (SMA) actuator by utilizing its self-sensing capabilities. This is based on the close to linear relation between the displacement of the wire length and the electric resistance within the wire.

When it comes to research for a project of this magnitude, it is important to broaden the views early. One seemingly ideal solution can prove not to be as relevant as the project progresses. At occasions as this it is beneficial to have alternative and supplementary articles to diversify the perspective and broaden the views of the person carrying out the project.

Tests were conducted to create datasets on which the polynomial fitted functions are based, for both the total voltage ( $V_{tot}$ ) and the displacement ( $D_{S,est}$ ). Additionally, because of a bias voltage from the capacitor in the active low-pass filter a function was derived to compensate for a non-linear amplification when the PWM signal was low. This resulted in an estimator of the displacement which consisted of a poor tracking ability with rapid and violent perturbations. However, the actual displacement ( $D_S$ ), in spite of the poor estimate, was able to track the set point to some degree. If the estimate had been filtered with a lower cutoff frequency, the performance would have improved, but the cutoff frequency was too high.

The conclusion drawn from this is that the main challenge lies with the measurement timing with the PWM signal. The optimal solution could be to implement a soft PWM program which always performs the analog read when the PWM signal is high. This means that the active LP-filter is redundant and can be removed. Thus, simplifying the circuit significantly.

However, this solution, if optimized, is highly applicable for actuating a robotic gripper in an end effector with tactile sensor feedback. Mainly because the reduced requirements in accuracy of the estimate ( $D_{S,est}$ ) since the tactile sensor will tell when the displacement has reached its set point.

## 6.1 Future work

To optimize the sensor-less control of the SMA actuator the measurement timing issue must be handled, either by implementing a soft PWM or another measurement technique. Additionally, more accurate mathematical models can advantageously be implemented to compensate for the dynamics of the actuator wire. This could be done by implementing a feedforward controller such that the tracking capabilities are improved. Lastly, to improve the responsiveness of the system the values for current through the wire should operate close to the recommended maximum values.

However, if the system is implemented in some form of prosthetics or other devices to interact with bionic-material such as human skin, it is important to isolate the actuator or bundle of actuators from contact because of the high temperature.

Further, by eventually implementing a tactile sensor for additional feedback with regards to grip force will give the system the possibility to detect when the object is in contact with the gripper. This means that the SMA actuator has reached its set point for the displacement. Additionally, the tactile sensors will have their own set points with regards to applied grip force. Though, this will be different depending on the object that is being picked up, but this will be decided by a different system designed in the project by SINTEF.

# References

- Arduino. Arduino uno specifications. On Arduinos homepage, May 2016. URL [<https://www.arduino.cc/en/main/arduinoBoardUno>].
- M. Cianchetti, A. Licofonte, M. Follador, F. Rogai, and C. Laschi. Bioinspired soft actuation system using shape memory alloys. In *Actuators*, volume 3, pages 226–244. Multidisciplinary Digital Publishing Institute, 2014.
- Digikey. Circuit schematic designer. Web designer, April 2016. URL [<http://www.digikey.com/schemeit/>].
- S. Dutta and F. Ghorbel. Differential hysteresis modeling of a shape memory alloy wire actuator. *IEEE/ASME Transactions on Mechatronics*, 10(2):189–197, apr 2005. doi: 10.1109/tmech.2005.844709. URL [<http://dx.doi.org/10.1109/TMECH.2005.844709>].
- Dynalloy. Technical characteristics of flexinol actuator wires. [<http://www.dynalloy.com/pdfs/TCF1140.pdf>], 2016. URL [<http://www.dynalloy.com/pdfs/TCF1140.pdf>]. Besøkt: [september 2015].
- T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Inc, 2011. ISBN 1119991498. URL [http://www.ebook.de/de/product/15427426/thor\\_i\\_fossen\\_handbook\\_of\\_marine\\_craft\\_hydrodynamics\\_and\\_motion\\_control.html](http://www.ebook.de/de/product/15427426/thor_i_fossen_handbook_of_marine_craft_hydrodynamics_and_motion_control.html).
- D. Grant and V. Hayward. Variable structure control of shape memory alloy actuators. *IEEE Control Syst. Mag.*, 17(3):80–88, jun 1997. doi: 10.1109/37.588180. URL [<http://dx.doi.org/10.1109/37.588180>].
- D. Hughes and J. T. Wen. Preisach modeling of piezoceramic and shape memory alloy hysteresis. *Smart materials and structures*, 6(3):287, 1997.
- JGraph. Flowchart maker & online diagram software. Web designer, May 2016. URL [<https://www.draw.io/>].
- B. Kim, S. Lee, J. Park, and J.-O. Park. Design and fabrication of a locomotive mechanism for capsule-type endoscopes using shape memory alloys (SMAs). *IEEE/ASME Transactions on Mechatronics*, 10(1):77–86, feb 2005. doi: 10.1109/tmech.2004.842222. URL [<http://dx.doi.org/10.1109/TMECH.2004.842222>].
- B. Kim, M. G. Lee, Y. P. Lee, Y. Kim, and G. Lee. An earthworm-like micro robot using shape memory alloy actuator. *Sensors and Actuators A: Physical*, 125(2):429–437, jan 2006. doi: 10.1016/j.sna.2005.05.004. URL <http://dx.doi.org/10.1016/j.sna.2005.05.004>.

- K. J. D. Laurentis and C. Mavroidis. Mechanical design of a shape memory alloy actuated prosthetic hand. *Technology and Health Care*, 10(2):91–106, 2002.
- J. Lee, M. Jin, and K. K. Ahn. Precise tracking control of shape memory alloy actuator systems using hyperbolic tangential sliding mode control with time delay estimation. *Mechatronics*, 23(3):310–317, apr 2013. doi: 10.1016/j.mechatronics.2013.01.005. URL [<http://dx.doi.org/10.1016/j.mechatronics.2013.01.005>].
- S.-H. Liu, T.-S. Huang, and J.-Y. Yen. Tracking control of shape-memory-alloy actuators based on self-sensing feedback and inverse hysteresis compensation. *Sensors*, 10(1):112–127, dec 2009. doi: 10.3390/s100100112. URL [<http://dx.doi.org/10.3390/s100100112>].
- N. Ma and G. Song. Control of shape memory alloy actuator using pulse width modulation. *Smart materials and structures*, 12(5):712, 2003.
- M. V. Mark W. Spong, Seth Hutchinson. *Robot Modeling and Control*. JOHN WILEY & SONS INC, 2005. ISBN 0471649902. URL [http://www.ebook.de/de/product/5124811/mark\\_w\\_spong\\_seth\\_hutchinson\\_m\\_vidyasagar\\_robot\\_modeling\\_and\\_control.html](http://www.ebook.de/de/product/5124811/mark_w_spong_seth_hutchinson_m_vidyasagar_robot_modeling_and_control.html).
- M. J. Mosley and C. Mavroidis. Experimental nonlinear dynamics of a shape memory alloy wire bundle actuator. *J. Dyn. Sys., Meas., Control*, 123(1):103, 2001. doi: 10.1115/1.1344243. URL <http://dx.doi.org/10.1115/1.1344243>.
- M. F. M. Naser and F. Ikhoulane. Characterization of the hysteresis duhem model. *IFAC Proceedings Volumes*, 46(12):29–34, 2013.
- National Instruments. Linx faq (frequently asked questions). Published on LabVIEW MakerHubs homepage, may 2016a. URL [[https://www.labviewmakerhub.com/doku.php?id=libraries:linx:faq&s\[\]=compiler#8](https://www.labviewmakerhub.com/doku.php?id=libraries:linx:faq&s[]=compiler#8)].
- National Instruments. How linx works. Published on LabVIEW MakerHubs homepage, may 2016b. URL [<https://www.labviewmakerhub.com/doku.php?id=learn:libraries:linx:misc:linx-internals>].
- P. Senthilkumar, G. N. Dayananda, M. Umopathy, and V. Shankar. Experimental evaluation of a shape memory alloy wire actuator with a modulated adaptive controller for position control. *Smart Mater. Struct.*, 21(1):015015, dec 2011. doi: 10.1088/0964-1726/21/1/015015. URL [<http://dx.doi.org/10.1088/0964-1726/21/1/015015>].
- L. Stirling, C.-H. Yu, J. Miller, E. Hawkes, R. Wood, E. Goldfield, and R. Nagpal. Applicability of shape memory alloy wire for an active, soft orthotic. *J. of Materi Eng and Perform*, 20(4-5):658–662, feb 2011. doi: 10.1007/s11665-011-9858-7. URL <http://dx.doi.org/10.1007/s11665-011-9858-7>.
- A. Visintin. Mathematical models of hysteresis. *The science of hysteresis*, 1: 1–114, 2006.



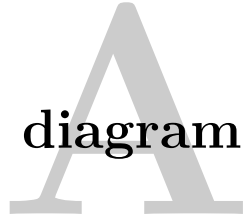
A. Visintin. *Differential models of hysteresis*, volume 111. Springer Science & Business Media, 2013.

E. W. Weisstein. "least squares fitting". from mathworld - a wolfram web resource. Published on Wolfram Web Resource, May 2016. URL [<http://mathworld.wolfram.com/LeastSquaresFitting.html>].



# Appendix

## Flow diagram



On the next page is the total flow diagram for the LabVIEW program. In the flow chart the colors represent a function in the code.



**Figure A.1:** Color explanation for figure A.2.

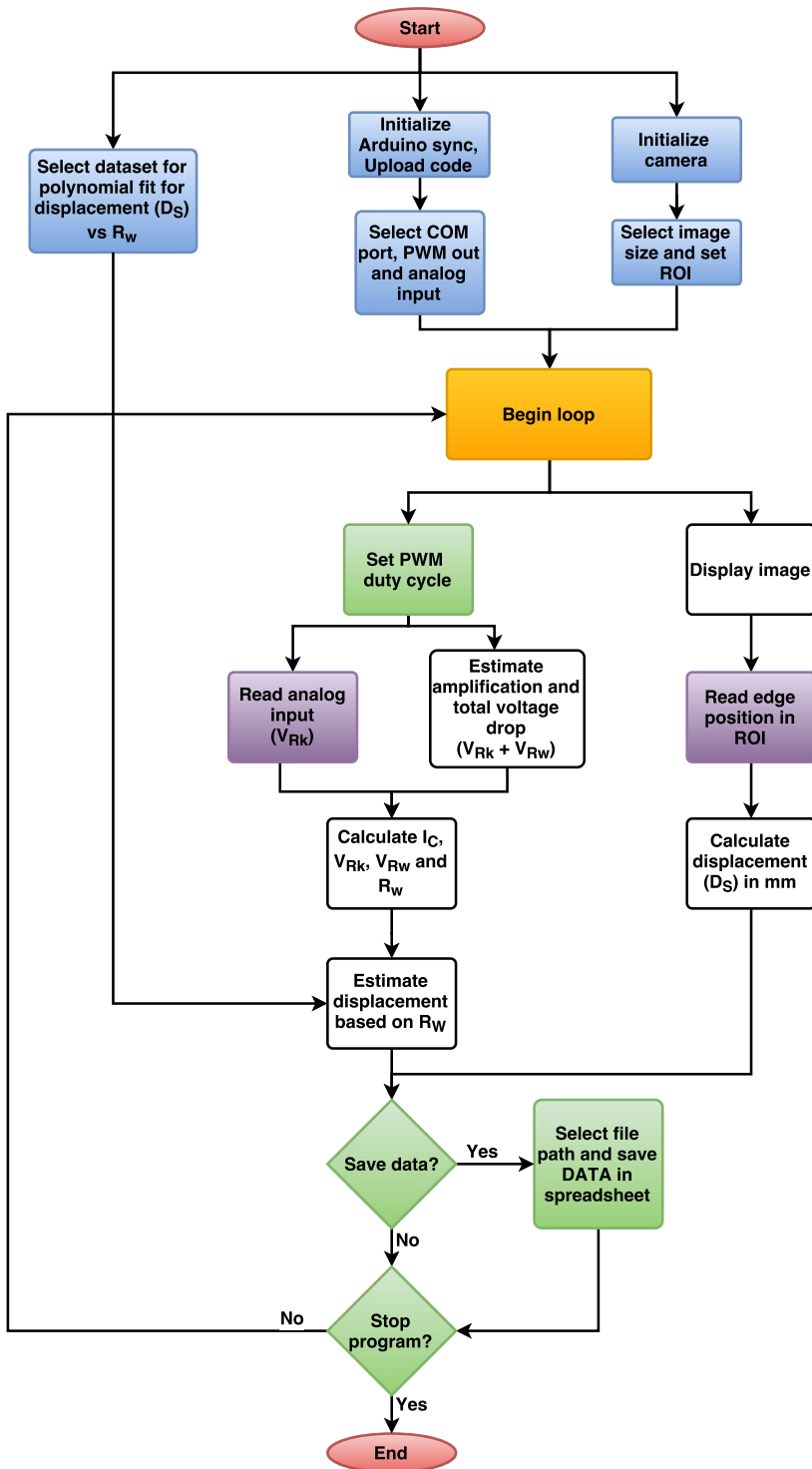
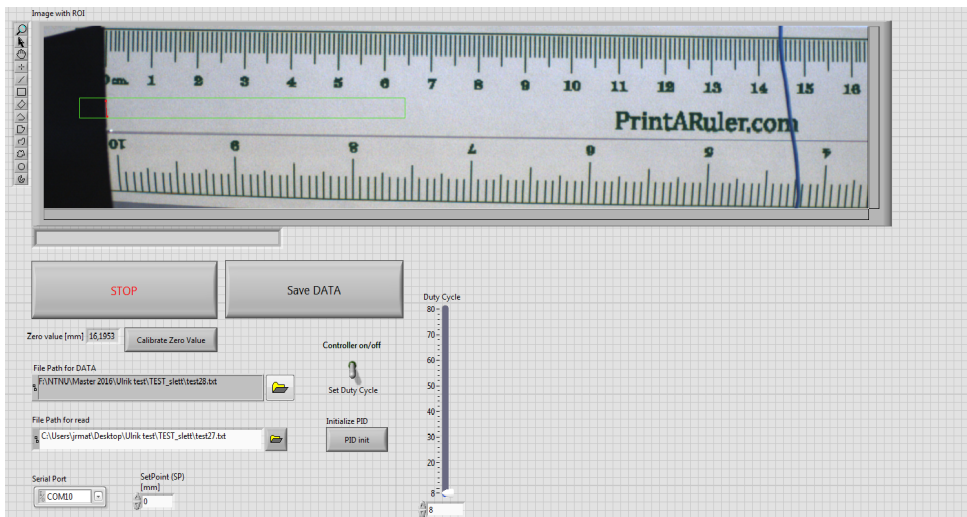


Figure A.2: Total flowdiagram of the LabVIEW code.

# Appendix B

## LabVIEW - Front panel

The front panel is divided into two sections. The area where the user actually can change paths and set point, press buttons and stop the program is called the *control section* and the area where all the parameters are plotted is called the *observation section*.



**Figure B.1:** Screenshot from the “*control section*” in the front panel of the LabVIEW code.

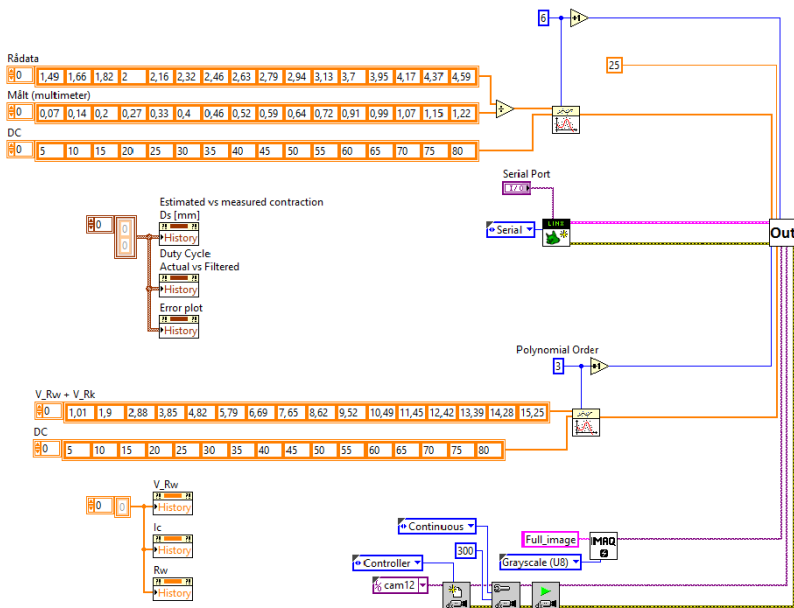


**Figure B.2:** Screenshot from the “*observation section*” in the front panel of the LabVIEW code.

# Appendix C

## LabVIEW - Block diagram

Figure C.1 is a code snip of the block diagram for the initialization of before the main loop in figure C.2 of the LabVIEW program.



**Figure C.1:** The bolck diagram of the LabVIEW code.

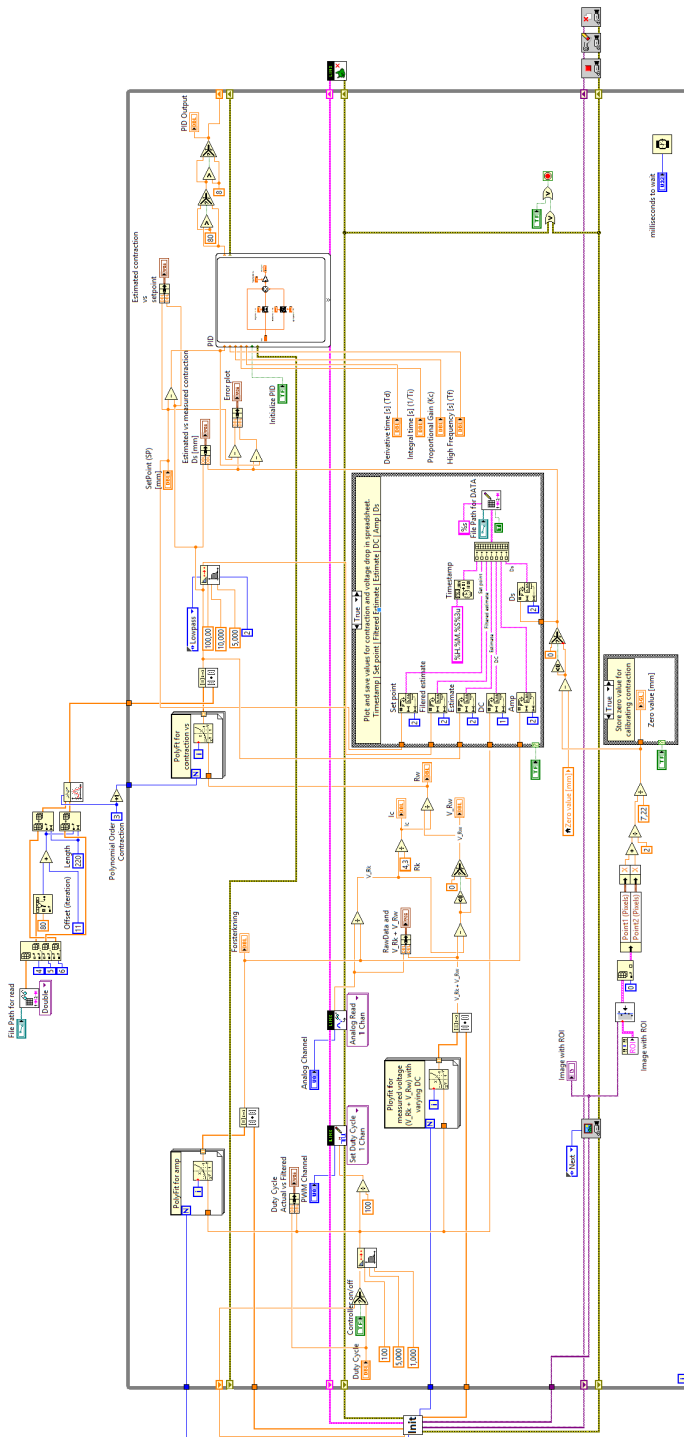


Figure C.2: The bolck diagram of the LabVIEW code.



# Appendix **D**

## Camera image

On the next page is a cropped image of the camera view with the ROI and the detected edge in the image.



**Figure D.1:** Camera image with ROI (green square) and detected edge (red line). The axes are added after the image is captured for reference.

# Appendix **E**

## Photos of the solution

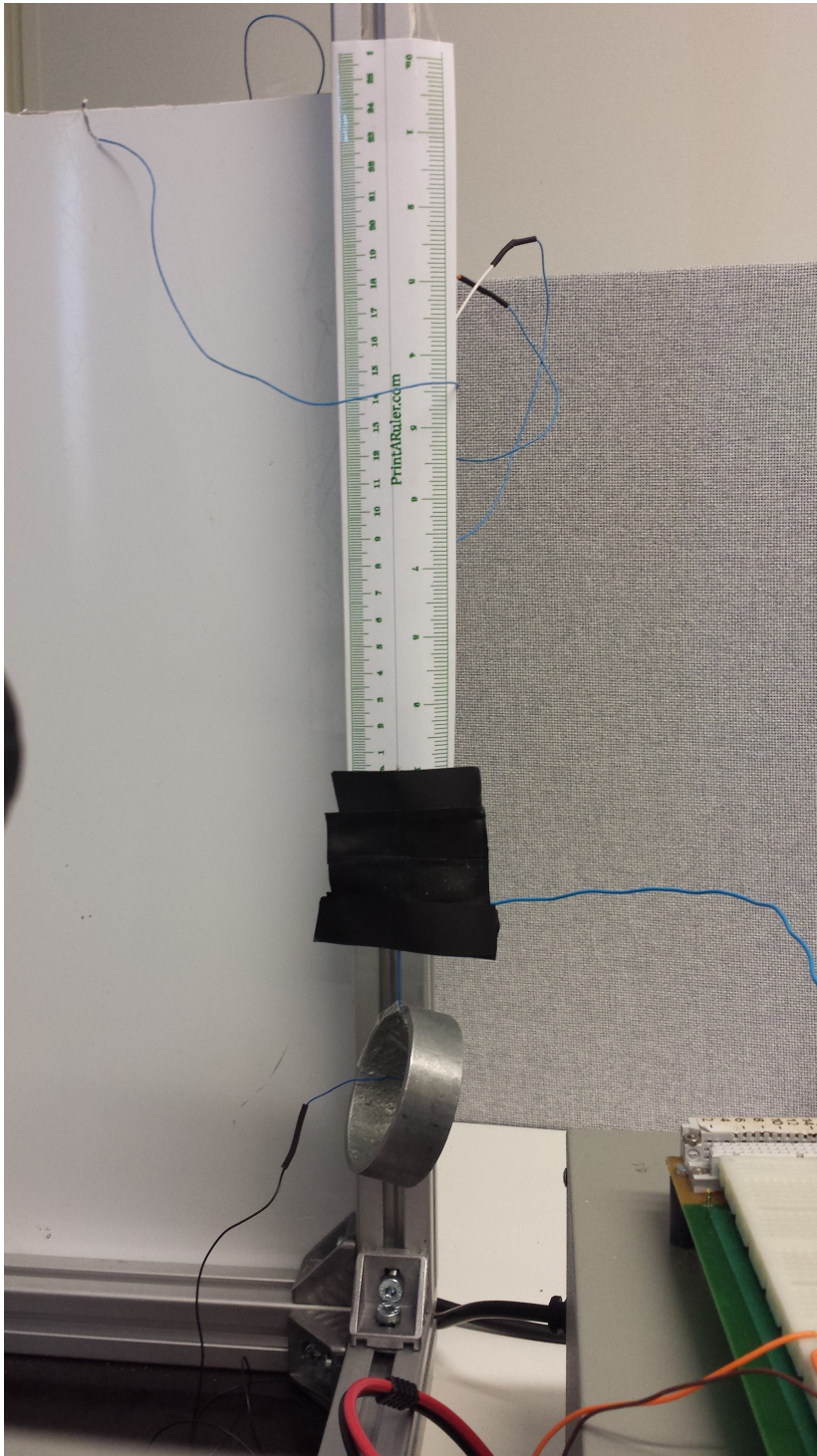
Following this appendix are images of the final solution. On the next page is an image of the physical rig of the experimental setup compared to the *CAD*-sketch, and after that is an image of the rig with the wire hanging in front of the camera with the tape used for the edge detection and a 96 *gram* weight attached to it.



**Figure E.1:** Image of the camera used to make the measurements of the actual displacement ( $D_S$ ).



**Figure E.2:** The actual test rig (to the left) compared to the *CAD*-sketch (to the right).



**Figure E.3:** Image of the wire with the tape for the edge detector and a 96 *gram* weight. View from the camera.