Modeling and Control of a TCPM

T.R. Robeerts

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Modeling and Control of a TCPM

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

T.R. Robeerts

April 16, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Twisted and coiled polymer muscles (TCPMs) show promise to function as artificial muscles, because of their lightweight, low cost, large contraction, and respectively low hysteresis. A TCPM contracts when it is heated and extends when it is cooled. Different modeling and controlling techniques have been implemented. Van der Weijde et al. [2019] implemented a self-sensing model that does not need large apparatus for measurements of force and deflection. The goal of this thesis is to design a force controller that works with this model. Parameter estimation of the self-sensing model is done. The fit of the model is not high enough for control. A first order black-box model is estimated and used instead. A P and PI controller is simulated and tested on the setup. The force oscillates around the reference value. This is because the actual model is of order 2. A D-action needs to be added to dampen the oscillations. The integral action reduces the max to min and vice versa input behavior. The model parameter differs for each TCPM. The controller parameters have to be adjusted for each TCPM. This is impractical in large-scale applications. Further research can be done into using model-free controllers.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors Prof.dr. R. Babuska and Prof.dr.ing. H. Vallery for their assistance during the writing of this thesis. I would like to thank Joost van der Weijde for all the information he has provided me.

Delft, University of Technology T.R. Robeerts
April 16, 2021

Chapter 1

Introduction

The interest in obtaining actuators with similar behavior and characteristics as biological muscles increases. Haines et al. [2014] propose an artificial muscle made of a twisted and coiled polymeric fiber. This is called a twisted coiled polymer muscle (TCPM). These TCPMs contract when heated and extend when cooled down. They have several advantages over other artificial muscles such as shape memory alloys. The advantages are its low weight, large contractions, high load-carrying capabilities, respectively low hysteresis compared to other types of artificial muscles, and low cost.

Different types of models and controllers have been researched and implemented to describe the TCPM and control its deflection and force. The most common model is a combination of a thermomechanical and thermoelectric model, such as Yip and Niemeyer [2015], Simeonov et al. [2018], Sutton et al. [2016], Masuya et al. [2018], Suzuki and Kamamichi [2018], and Wu and Tadesse [2017]. Open and closed-loop controllers have been implemented by for example Jafarzadeh et al. [2018], Yip and Niemeyer [2017], Yip and Niemeyer [2017], Masuya et al. [2018], Zhang et al. [2017], Sutton et al. [2016], Arakawa et al. [2016]. Each TCPM is produced differently. The most applied form of heating is Joule heating. Other forms are using hot air (Abbas and Zhao [2017] and Cherubini et al. [2015]) and water (Yip and Niemeyer [2017] and Wu et al. [2015]. The most used cooling process is air, either moving or still. Others have used water (Yip and Niemeyer [2017] and Wu et al. [2015]).

The current controllers need large apparatus to determine the force and strength. This is not practical in robotic applications. An example is the assistive wrist orthosis designed by Sutton et al. [2016]. This orthosis uses TCPMs as actuators of which the force needs to be measured and adjusted. Van der Weijde et al. [2019] implemented a self-sensing model such that large apparatus for measurements are not necessary anymore. A small device called the muscle drive (MD) measures the decay time, a measure of inductance. The decay time can be used to determine the deflection and the force of the TCPM. The goal of this thesis is to implement a controller to control the force of a TCPM that works with the self-sensing model.

This goal is not achieved since the self-sensing model did not obtain the desired results (see Chapter 3). A black-box model that relates the input to the output is used instead. It is shown that a TCPM can be controlled by using a P or PI-controller on a first order black-box model. A disadvantage is that each produced TCPM behaves differently. This means that the modeling process and controller design needs to be repeated for each TCPM. This is not achievable for large-scale applications.

The working principle and production process of a TCPM is explained in Chapter 2. In this chapter, the experimental setup is also explained. The self-sensing model and its parameter estimation are discussed in Chapter 3. The modeling of the TCPM without the self-sensing model is explained in Chapter 4. The controller design is discussed in Chapter 5. Last, the conclusions and further research recommendations are discussed in Chapter 6.

Chapter 2

Twisted and coiled polymer muscles

An understanding of the TCPM is necessary to be able to design a model and a controller. This chapter explains the working principle of a polymer muscle, its production process, and the setup used. The working principle and the general production process of a twisted and coiled polymer muscle are discussed in Section 2-1. The used production process of the TCPMs and the experimental setup are discussed in Section 2-2 and Section 2-3.

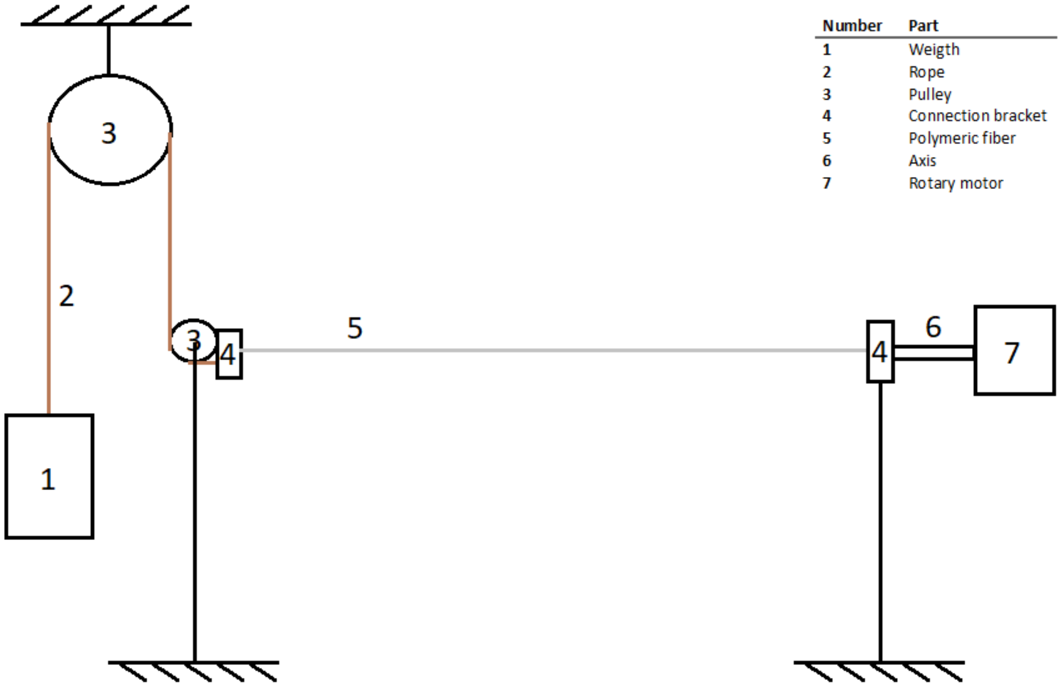
2-1 Working principle and general production process

Polymeric fibers contract when it is heated because it has a negative thermal expansion in axial direction. Twisting and coiling the fiber increases this strain. The increase depends on parameters such as the polymeric material, its dimensions, the load during twisting, the number of applied twists, and the coil diameter (Van Der Weijde et al. [2016]).

Polymeric fibers consist of chains. The polymeric fiber is in an amorphous phase, which means that most of its chains are not arranged in a pattern. These chains are bridged by crystalline bridges, which contain chains arranged in a pattern. These crystalline regions enable small thermal expansion coefficients. The fiber undergoes the process of cold drawn tension. This aligns the fiber in the direction of the drawing and ensures larger contractions and expansions because more crystalline regions are created. The fiber obtained from this process is called the precursor fiber and is the fiber available in shops, e.g. fishing line and sewing thread. The precursor fiber is twisted, creating helically oriented chains (Shafari and Li [2015]). The load used for twisting the fiber is crucial. Too large weight will break the fiber, while too little weight will snarl the fiber. The helically oriented chains contract during twisting, which leads to a radial expansion of the twisted fiber. This generates a torque in the opposite direction of the twist. The twisted fiber can function as a torsional muscle (Haines et al. [2014]). Coiling is needed to change the function of the fiber back to an actuator in axial direction. Two different processes can be used to obtain the coiling: twist-insertion by continuing twisting the fiber and thermal annealing of a twisted fiber coiled around a mandrel. Twist-insertion leads to auto-coiling. The twisted fiber will coil automatically when twist insertion is continued after the maximum number of twists of the fiber has been reached. A down-side of this technique is that the fiber can untwist, especially under load (Haines et al. [2014]). Thermal annealing of a twisted fiber coiled around a mandrel does not exhibit this disadvantage. The twisted fiber is wrapped around a mandrel, which leads to a spring-shaped muscle. Thermal annealing ensures that the shape remains after removing the mandrel. Thermal annealing is the process of heating the TCPM for a certain period of time with a temperature higher than the maximum operating temperature, but lower than the melting temperature of the polymeric fiber (Shafari and Li [2015]). Coiling the TCPM around a mandrel can be done in two directions. The muscle will contract when heated if the coiling is performed in the same direction as the twist. The TCPM will extend when heated if the coiling is done in the opposite direction as the twist. Haines et al. [2014] call this respectively homochiral and heterochiral.

2-2 Used production process

A homochiral TCPM is produced for the experiments in this thesis. The precursor fiber is made of nylon and has a diameter of 0.8 mm. A schematic overview of the setup used to twist the fiber is given in Figure 2-1.



Number	Part
1	Weigth
2	Rope
3	Pulley
4	Connection bracket
5	Polymeric fiber
6	Axis
7	Rotary motor

Figure 2-1: Schematic overview of the setup used to twist the polymeric fiber.

The nylon fiber is twisted together with a constantan wire with a diameter of 0.3 mm. The constantan wire is added to be able to heat the TCPM via Joule heating. A constantan wire is used because its resistance is almost constant with temperature. The advantage is that the deflection estimator in Section 3-1-1 can be simplified. The twist is inserted until auto-coiling starts as shown in Figure 2-2a. Figure 2-2b shows the twisted fiber.

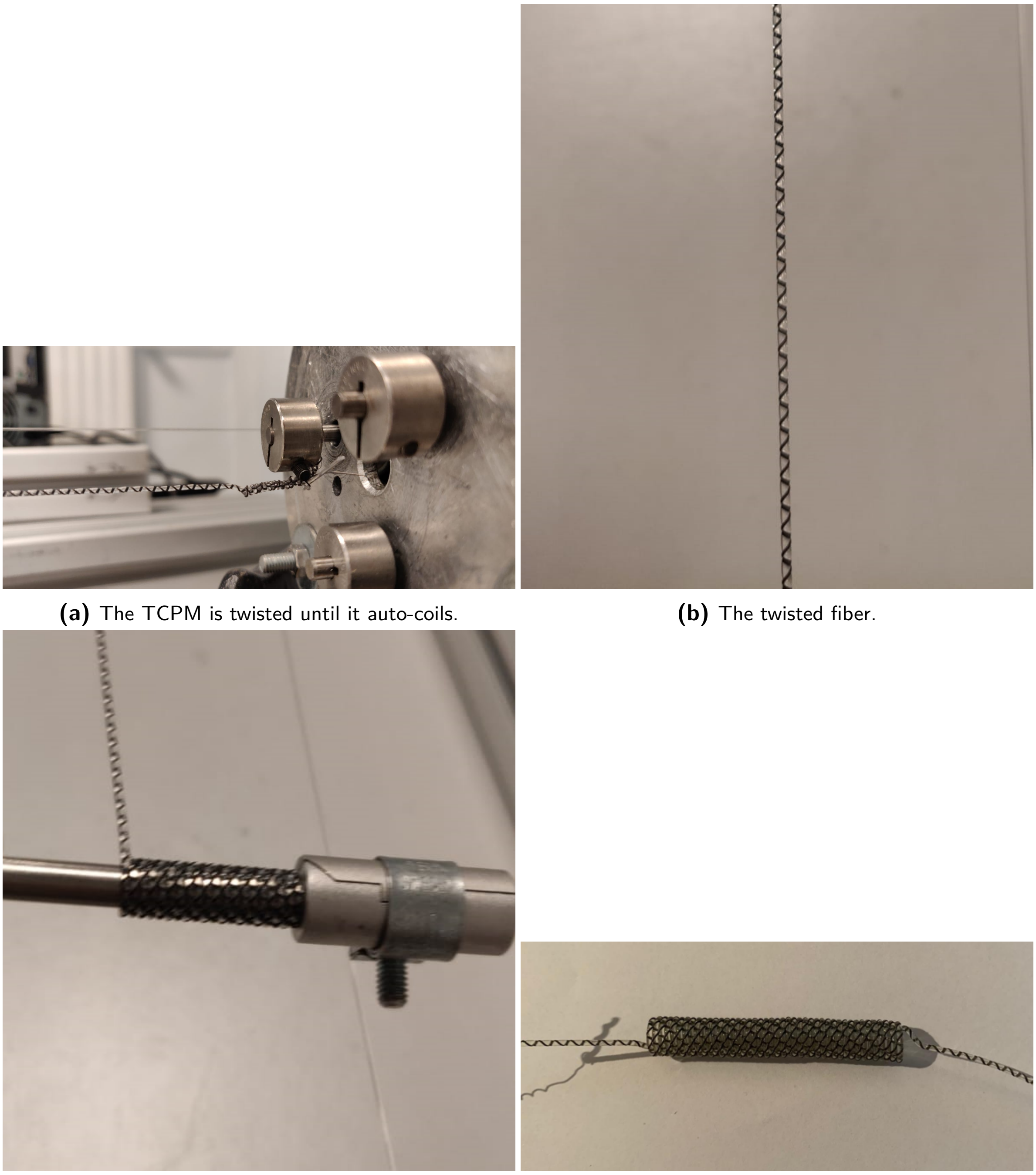
Coiling is done by wrapping the twisted fiber around a mandrel with a diameter of 5 mm and a length of 50 mm. This process is shown in Figure 2-2c. The number of obtained windings is 46. The weight used during twist insertion is approximately 3.00 N. The process of thermal annealing is applied to ensure that the shape of the TCPM remains after removing the mandrel. The muscle is placed in an oven with a temperature of 165 °C for an hour. The muscle is then cooled and removed from the mandrel. The resulting muscle can be seen in Figure 2-2d. An overview of this data is given in Table 2-1. Jumper heads are soldered at the ends of the constantan wire. A voltage supplier can be connected to the jumper heads for Joule heating. Eye terminals are mounted on the ends of the nylon fiber to be able to use the TCPM in the setup.

Table 2-1: Overview of the TCPM properties and the production properties.

Property	Value
Fiber material	Nylon
Fiber diameter	0.8 mm
Resistance wire material	Constantan
Resistance wire diameter	0.3 mm
Coil diameter	5 mm
Coil length	50 mm
Number of windings	46
Load of twist insertion	≈ 3.00 N
Thermal annealing temperature	165 °C
Thermal annealing time	1 hr

2-3 Experimental setup

An experimental setup is needed to produce input and output data for model design and to evaluate the performances of the controllers. The produced TCPM is mounted onto a loadcell in a Universal Testing Machine (UTM). This apparatus, the Mark10 ESM303, applies and measures the deflection and force on the TCPM. It has a resolution of 0.02 mm. The load cell of the UTM is a Mark10 M5-05 Force Gauge and has a resolution of 0.5 mN. A cooling fan is put in front of the TCPM holder to quicken the cooling process. An image of the setup is given in Figure 2-3a. The UTM is controlled via a Python script written by Joost van der Weijde and adapted to the purposes of this thesis. A Muscle Drive (MD) is used to control the process of Joule heating (Van der Weijde et al. [2019]). This module is designed by Joost van der Weijde and also uses a Python script. The MD can be used to send the duty cycle needed for a certain voltage to heat or stop heating the TCPM. It can also be used to measure the inductance of the muscle. The inductance can be used to determine the deflection and force of the TCPM. This is called self-sensing, see Chapter 3. An advantage is that no large equipment is needed in practical applications to determine the deflection and force of a TCPM. An image of the MD is given in Figure 2-3b The MD is connected to a voltage supplier.



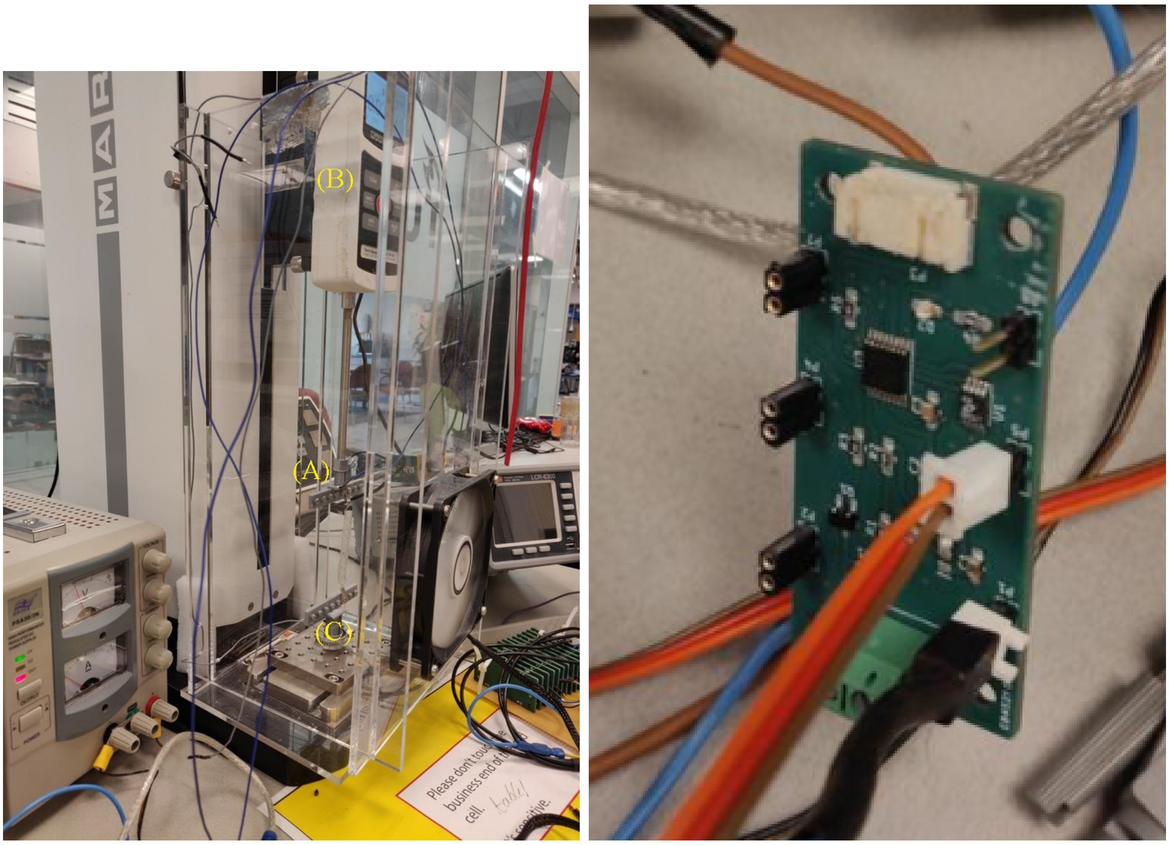
(a) The TCPM is twisted until it auto-coils.

(b) The twisted fiber.

(c) The twisted fiber is wrapped around a mandrel.

(d) The TCPM after the production is finished.

Figure 2-2: Production process of a TCPM.



(a) The UTM MARK10 ESM303 with the load cell MARK10 M5-05 and the cooling fan.

(b) The muscle drive that controls the process of Joule heating.

Figure 2-3: Experimental setup.

The TCPM needs to be connected to the experimental setup and the experimental setup needs to be calibrated. Figure 2-3a is labeled to explain this more clearly. One end of the TCPM is connected to the movable arm of the UTM (A). The force measured by the load gauge (B) is set to 0 when the TCPM is suspended. The loose end of the TCPM is then connected to the bottom side of the TCPM (C). This side is fixed. The TCPM is then stretched such that it just starts to be under tension with a force of 0.2 N. The measured position of the UTM is calibrated to 0.

The TCPMs produced with the production process described in Section 2-2 are used on this experimental setup to determine the model parameters, see Chapter 3 and Chapter 4, and to evaluate the designed controllers, see Chapter 5.

Chapter 3

Self-sensing model

Van der Weijde et al. [2019] implemented a self-sensing model. This model can estimate the force and the deflection of a TCPM using its decay time, a measure of the inductance. The advantage is that large apparatus to measure the position and force, such as the UTM, can be replaced with a smaller one, for example, the MD. This makes it more suitable for practical implementations. First, the model designed by Van der Weijde et al. [2019] is discussed. Then, how the parameters of this model are estimated. Last, its performance on the setup is analyzed.

3-1 The self-sensing model

The self-sensing model consists of three submodels: a deflection estimator, a temperature model, and a force model. An overview is given in Figure 3-1. The deflection estimator estimates deflection ($\hat{x}$) based on the decay time (L) measured by the MD. The temperature model estimates the contribution of temperature to the force (T) based on the power input send to the TCPM. This temperature is not the actual temperature in the TCPM, but it explains the contribution. The force model uses the estimate of the deflection and the contribution of the temperature to estimate the force of the TCPM ($\hat{F}$). Each submodel will now be discussed in further detail.

3-1-1 The deflection estimator

The deflection estimator describes the relation between a measure of inductance (the decay time) and the deflection. The inductance of a coil is in its simplest form given by (Serway and Jewett [2013]):

$$L(\Delta x) = \mu_0 \frac{N^2}{l} \pi r^2 \tag{3-1}$$

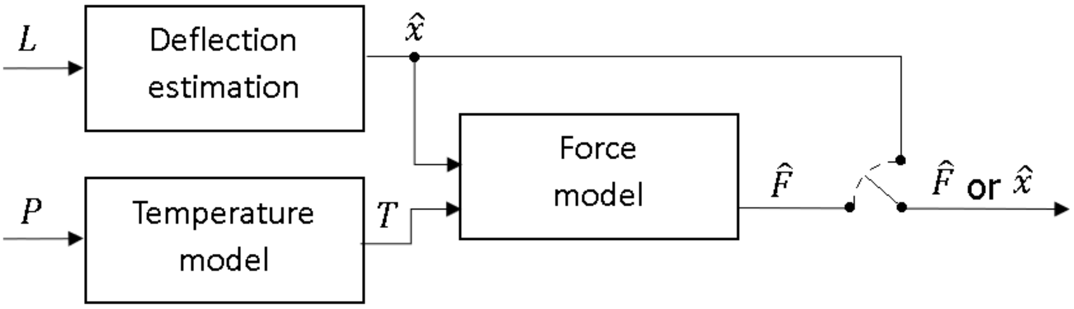


Figure 3-1: Schematic overview of the self-sensing model.

where μ_0 is the magnetic permeability, N the number of windings in the coil, l the length of the coil, and r the radius of the windings. This equation is not accurate enough for springs with finite lengths, a distance between the windings, and a round wire. The TCPM has all of these three characteristics. Corrections on the equation are therefore necessary. Van Der Weijde et al. [2015] researches various corrections and proposes the simplified model:

$$L = \frac{\lambda_x}{l} + \lambda_0 \tag{3-2}$$

where λ_x is the scaling parameter and λ_0 the offset. Its values can be determined with parameter estimation. This model fits the general shape of the curve of inductance and the parameters can be estimated with a limited data set. This model is still not accurate enough for the general TCPM, because of the changes in behavior due to temperature changes. The TCPM is heated and this changes the material properties and the geometry. Van Der Weijde et al. [2016] adds a temperature term to the model, because an increase in temperature leads to an offset in the inductance. The model for the inductance becomes:

$$L(t) = \frac{\lambda_x}{x(t) + \lambda_l} + \lambda_T T(t) + \lambda_0 \tag{3-3}$$

where λ_T is the fitted parameter for the temperature contribution. The length is changed to an addition of $x(t)$ and λ_l, where $x(t)$ is the deflection function and λ_l a parameter that relates to the initial length of the TCPM. Note that T and L do not represent the physical temperature and inductance, but a measurement proportionally related. The measure of the inductance determined with the MD is the decay time.

In the used deflection estimator the added temperature term is neglected because a metal wire with a resistance almost constant with varying temperature is used, namely constantan (see Section 2-2). This simplifies the model to the equation in (3-2). Rewriting this equation leads to the deflection estimator of the self-sensing model (Van der Weijde et al. [2019]:

$$x(t) = \frac{\lambda_x - \lambda_l(L(t) - \lambda_0)}{L(t) - \lambda_0} \tag{3-4}$$

3-1-2 The temperature model

The temperature model describes the relationship between the power input and the contribution of the temperature to the force. It is assumed that the TCPM is heated homogeneously. The duration of this is neglected. Van der Weijde et al. [2019] states the relationship between the contribution of the temperature and the power ($P(t)$) is of first order:

$$\dot{T}(t) = \kappa_P P(t) - \kappa_c T(t) \tag{3-5}$$

where κ_P represents the coefficient of conductive heating and κ_c the coefficient of convective cooling. The relation between the power input and the duty cycle is added to this model since the duty cycle is used as input for control in this thesis instead of the power. The relation between the duty cycle ($D(t)$) and the power is (Van der Weijde et al. [2019]):

$$P(t) = D(t)^2 R_m \Big(\frac{U_b}{R_b}\Big) \tag{3-6}$$

where R_m is the resistance measured at the two ends of TCPM, R_b the resistance measured at the connectors to the power supply, and U_b the voltage applied at the connectors. The duty cycle is directly saved by the MD. Determining the power also needs the resistances of the TCPM. This is a measured value and has inaccuracies. The value of the resistance changes when the resistance is measured at a slightly higher or lower position.

3-1-3 The force model

The force model describes the relationship between the estimated force and the estimated deflection and temperature contribution. Van der Weijde et al. [2019] combines a Standard Linear Solid model (Roylance [2001]) with a temperature term as follows:

$$F(t) = F_l(t) + T(t) + F_o(t) \tag{3-7}$$

where F_0 is a force offset and F_l is the contribution by the SLS model. F_l can be determined with:

$$\dot{F_l}(t) = \frac{-k_2}{c} F_l(t) + \frac{k_1 k_2}{c} x(t) + (k_1 + k_2)\dot{x}(t) \tag{3-8}$$

where $k1$ and $k2$ are stiffnesses and c is the damping coefficient. These parameters and F_0 can be fitted using input and output data.

3-2 Parameter estimation

The parameters of the self-sensing model need to be determined to predict the deflection and the force of the TCPM for control. For each TCPM these parameters differ. This means that for each used TCPM the parameters have to be determined. The process of parameter estimation is described here. First, the input signal is described. Then, data processing is needed to perform parameter estimation. Then, the method of parameter estimation itself. Last, the results are discussed.

3-2-1 Input signal

First the muscle is trained by applying a 600 s multi-sine signal to the deflection and power. The duty cycle and deflection are varied as input for the identification of the parameters of the self-sensing model. The same signal construction is used for both. First, the muscle is warmed up by applying a 250 s multi-sine. Then a 200 s multi-sine followed by a 200 s random-step signal with some rest in between is used for identification data. Last, a multi-sine of 100 s and a random-step signal of 120 s with some rest in between is applied for validation data. This is based on the work done by Van der Weijde et al. [2019]. The multi-sine is implemented with the following equation:

$$m(t) = a_0 + \sum_{i=1}^{N} a_i \sin(2\pi f_i t + \phi_i) \tag{3-9}$$

where N is the number of components in the multi-sine, a_0 the signal offset, a_i the amplitude of the i^{th} component, f_i its frequency, and ϕ_i its phase. The phases are determined with:

$$\phi_i = \phi_0 - \frac{\pi i^2}{N} \tag{3-10}$$

where ϕ_0 is a phase offset. The used frequency interval is $[10^{-2.4}, 10^{-1.1}]$ Hz which is spaced equally with N. N is set to 7. The random-step signal is created using the following equation:

$$g(t) = b_0 + \sum_{i=1}^{H} b_i h(t - \tau_i) \tag{3-11}$$

where H is the number of steps, b_0 the signal offset, b_i the height of the i^{th} step, τ_i its step time, and h the Heaviside step function. The step times are determined with a random generator. This random generator is similar to the construction of step times for generalized binary noise (Tulleken [1990]). The step sizes are sampled from the uniform distribution [0, $0.25(g_{\max} - g_{\min})$], where $g_{\max}$ and $g_{\min}$ represent the upper and lower limits of the power respectively. To prevent overheating the maximum of the duty cycle is set to 85%. The MD requires a minimum duty cycle of 15% to measure accurately. A virtual coin toss determines in which direction the step goes. The direction is made opposite when this means that the step size exceeds the maximum or is lower than the minimum value. The signal is scaled so that it includes the upper and lower limits of the deflection and the duty cycle.

The input signals used for the estimation of the parameters in the self-sensing model is given in Figure 3-2. It can be seen that during the run the duty cycle does not go below its minimum of 15% or above its maximum of 85%. The minimum and maximum used for the deflection is set to 10 mm and 30 mm to avoid nonlinear behavior caused by the touching of coils. The first 500 s are used for identification. The rest of the data is used for validation.

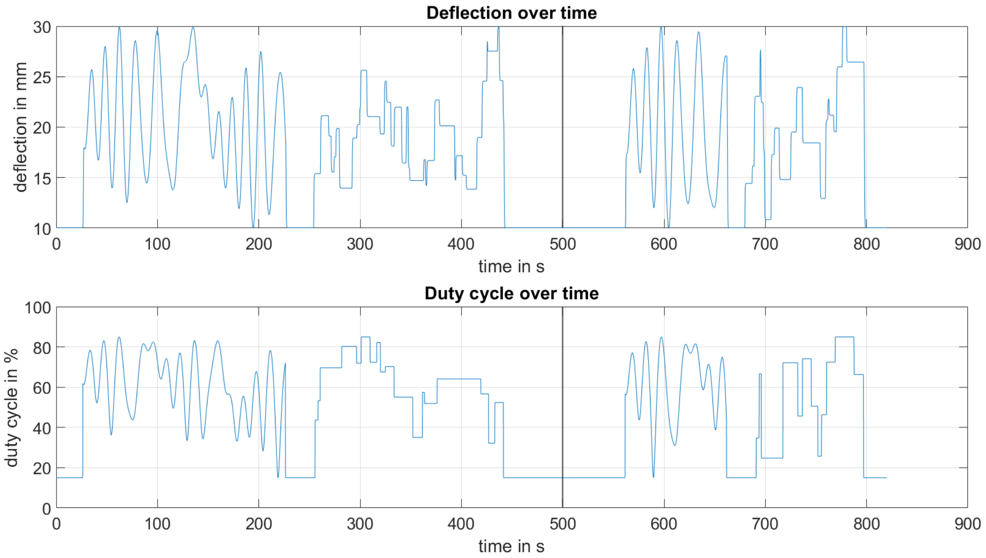


Figure 3-2: The duty cycle and deflection used as input signal for the identification of the parameters of the self-sensing model.

3-2-2 Data processing

The input and output data need to be processed and filtered before it can be used in the parameter estimation. The data is processed in the same way as Van der Weijde et al. [2019] implemented. The instants of time on which the UTM measures the force and the MD saves the duty cycle are not equal. The input and output data are resampled to 16 Hz. A 2-sample moving average filter and a 15-sample median filter is applied to the data to avoid spikes. The measured force is also filtered with a Butterworth low-pass filter. The amount of ripple allowed in the passband is set to 0.25 dB and the attenuation of the stopband is set to 0.75 dB. The example of input given in Figure 3-2 is already resampled. The resampled and filtered measured force is given in Figure 3-3.

3-2-3 Estimation of the parameters

The parameter estimation of the self-sensing model requires the values for R_m, R_b, and U_b. The values for two TCPMs are given in Table 3-1. The resistances are lower than the values obtained by Van der Weijde et al. [2019]. This shows that the production process by hand influences the resistance of the constantan wire.

First, κ_P, κ_c, F_o, c, k_1 en k_2 are identified. These are the parameters in the temperature and force model. The squared error between the measured and estimated force is minimized.

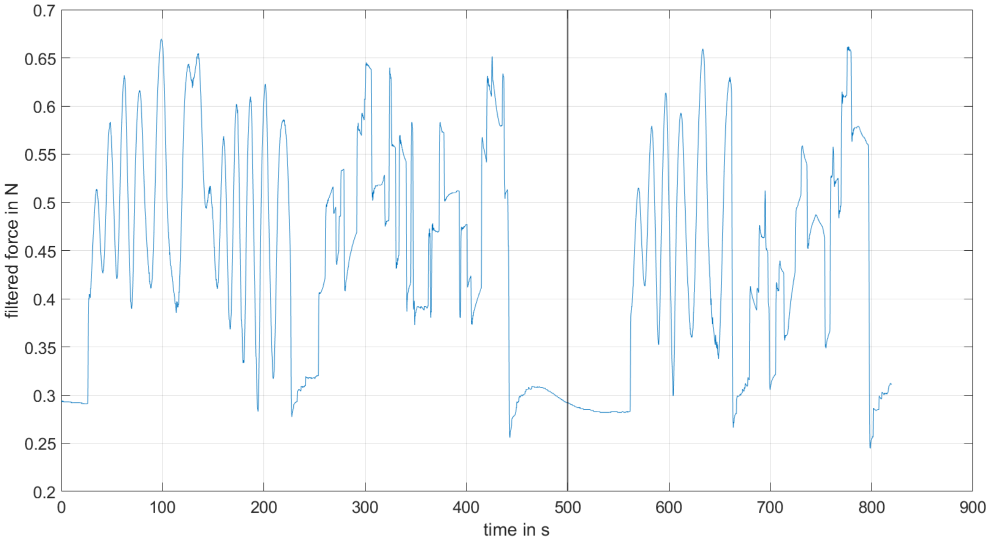


Figure 3-3: The duty cycle used as input signal for model estimation and an example of the obtained filtered measured force.

Table 3-1: The resistances of the TCPM and the applied voltage.

Property	Value TCPM1	Value TCPM2	Value Van der Weijde et al. [2019]
R_m	$9.60\,\Omega$	$9.82\,\Omega$	$10.18\,\Omega$
R_b	$9.96\,\Omega$	$10.22\,\Omega$	$10.75\,\Omega$
U_b	$7\,\text{V}$	$7\,\text{V}$	$7\,\text{V}$

The measured force is obtained with the UTM and the estimated force is obtained with the temperature and force model. This optimization is done by using MATLAB's nonlinear least-squares optimization. The Levenberg-Marquardt algorithm (LMA) is used for this. The algorithm is shown in Appendix B-2. Then, the parameters of the deflection estimator, λ_x, λ_l, and λ_0, are estimated similarly by minimizing the squared error between the applied deflection and the estimated deflection. The estimated deflection is obtained with the deflection estimator. The obtained parameters can differ for each try of parameter estimation, because of the used optimizing algorithm. The obtained parameters are given in Table 3-2. Parameters for two muscles and the parameters of Van Der Weijde are given to show that these parameters differ for each TCPM, but are mostly in the same order of magnitude.

The analysis of the estimation error of the deflection and the force is done in a similar way as Van der Weijde et al. [2019]. The Root Mean Square Error (RMSE) is used to quantify the estimation error. The quality of the fit is also checked by analyzing the R^2 value. This is given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3-12}$$

where n is the number of data point, y_i is the data itself, $\bar{y}$ is their mean, and f_i the estimates. The fit of a TCPM is shown in Figure 3-4. The applied power is determined by the duty

Table 3-2: The fitted parameters of the self-sensing model.

Parameter	Value TCPM 1	Value TCPM 2	Value Van der Weijde et al. [2019]
λ_x	6.29	2.64	2.81
λ_l	19.9	21.9	28.8
λ_0	0.389	0.477	0.433
κ_p	$2.9 \cdot 10^{-3}$	$5.2 \cdot 10^{-3}$	$7.2 \cdot 10^{-3}$
κ_c	$102.1 \cdot 10^{-3}$	$142.8 \cdot 10^{-3}$	$131.6 \cdot 10^{-3}$
k_1	$15.3 \cdot 10^{-3}$	$8.9 \cdot 10^{-3}$	$10.8 \cdot 10^{-3}$
k_2	$5.4 \cdot 10^{-3}$	$2.1 \cdot 10^{-3}$	$2.7 \cdot 10^{-3}$
c	$7.1 \cdot 10^{-3}$	$3.5 \cdot 10^{-3}$	$4.3 \cdot 10^{-3}$
F_0	$141.4 \cdot 10^{-3}$	$82.7 \cdot 10^{-3}$	$106.9 \cdot 10^{-3}$

cycle, see Equation (3-6). The fit of the other TCPM is given in Appendix A-1.

The first thing to notice is that the estimate of the deflection is not accurate as the fit obtained by Van der Weijde et al. [2019]. This is validated by the values of the RMSE and the R^2 value given in Table 3-3. The deflection estimate is not accurate in the first few seconds in contrast to what Van der Weijde et al. [2019] obtained. The deflection estimate increases in accuracy when time continues, but is still not as accurate as the estimates obtained by Van der Weijde et al. [2019]. The estimate of the force is more accurate. The estimate of the deflection plays a role in the force model. The error in the estimate of the deflection can cause problems for the estimate of the force on the setup.

Table 3-3: The R^2 and RMSE values for the deflection and force.

	TCPM 1				Van der Weijde et al. [2019]			
	Identification		Validation		Identification		Validation	
	R^2	RMSE	R^2	RMSE	$R^{@}$	RMSE	R^2	RMSE
x	0.57	3.40 mm	0.71	3.09 mm	0.96	0.97 mm	1.00	0.39 mm
F	0.91	27.8 mN	0.91	30.0 mN	0.97	12.8 mN	0.98	12.7 mN

3-2-4 Results

The accuracy of the self-sensing model is tested on the setup to see whether the model can be used in the control of the TCPM. The UTM varies the position and force by using a combination of step signals, sinusoidal signals, and constant signals. The self-sensing model with the obtained parameters is used to calculate what the force and the position would be. The results are shown in Figure 3-5.

The estimate of the deflection is more accurate than the estimate of the force. This is the opposite than what is obtained in the identification and validation. The position estimate is used in the force model increasing the inaccuracies caused by the error in the deflection estimate. The RMSE and the R^2 value are determined to quantify the accuracy of the self-sensing model. These values are given in Table 3-4.

The R^2 values are negative. This means that the estimated data does not follow the trend of the measured data. The RMSE values for the force are also a lot higher than in the

Table 3-4: The RMSE and R^2 value for the experimental data.

	R^2	RMSE
x	-4.358	5.88 mm
F	-0.956	113.9 mN

identification and validation data. The accuracy of the force estimate is not high enough for proper control and the reference value will not be obtained. At constant force, the self-sensing underestimates the force. It also underestimates the rate of change in the force. Therefore it is decided not to include the self-sensing model in the force-control of the TCPM.

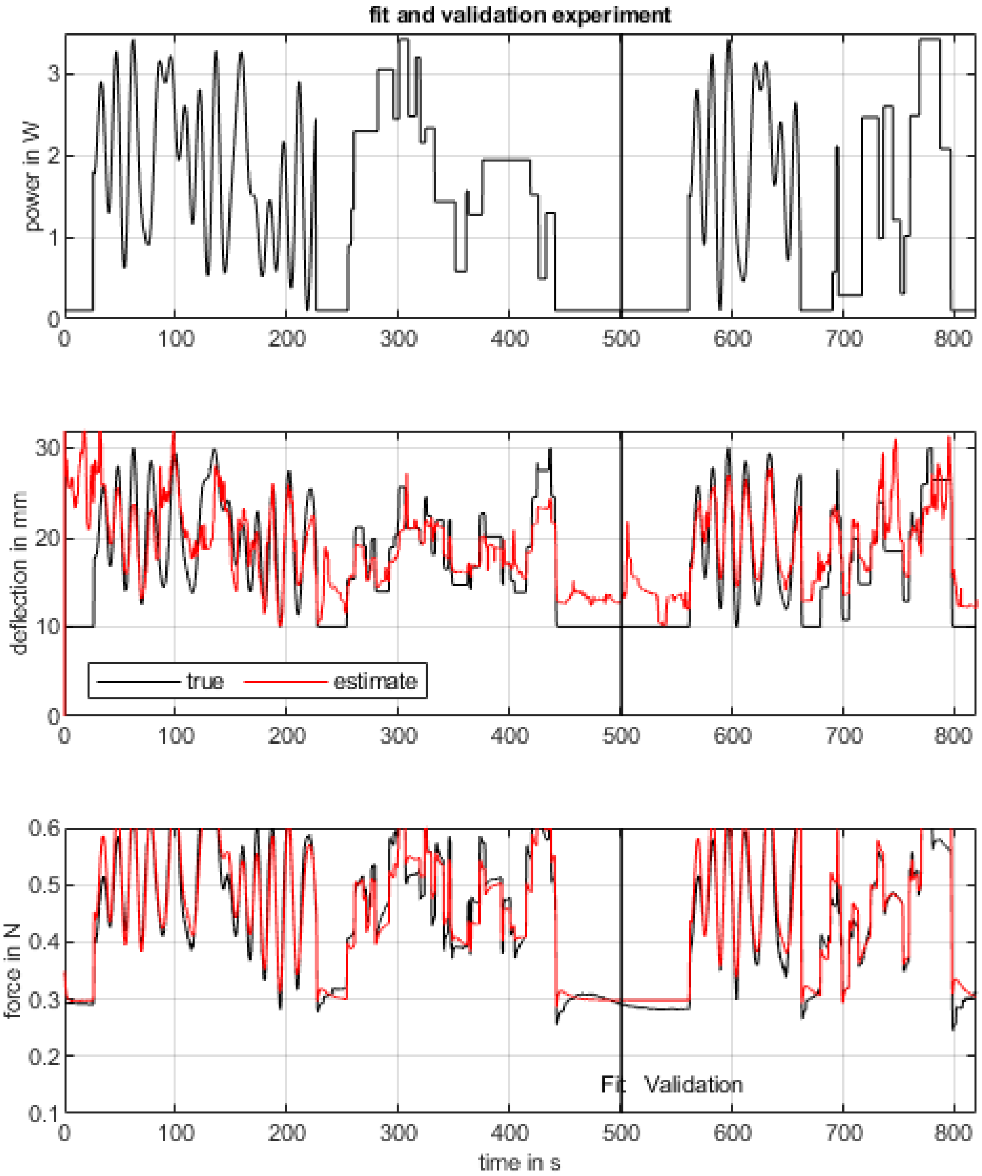
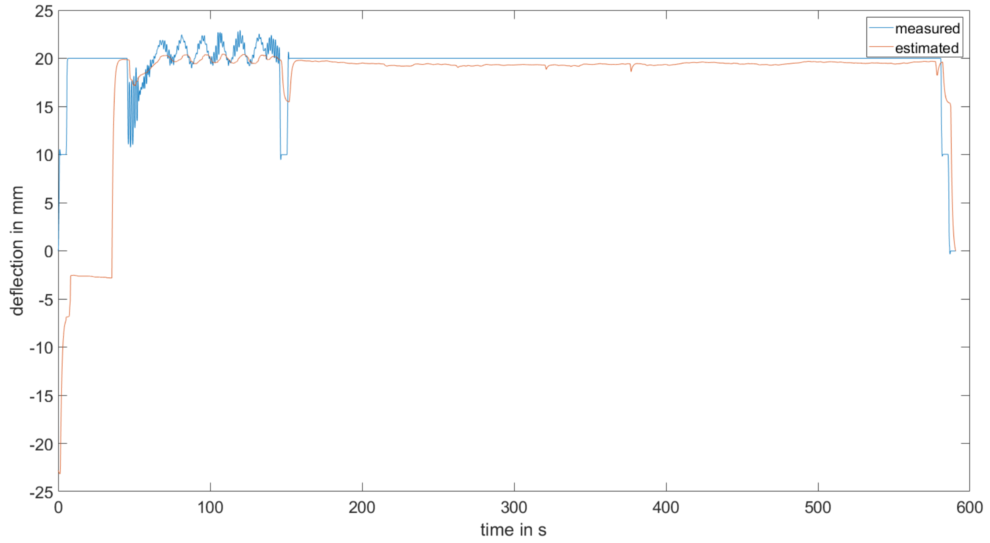
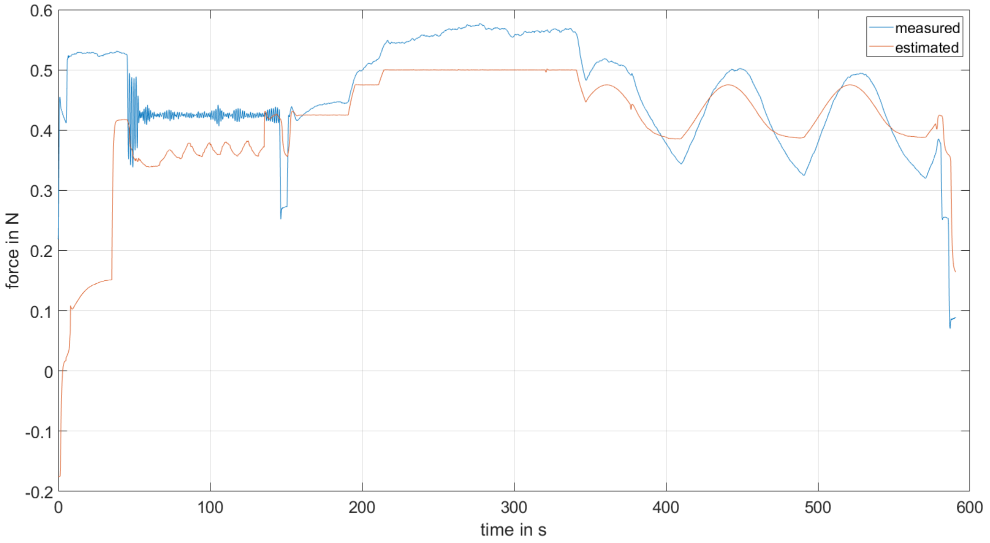


Figure 3-4: The top figure shows the applied power over time. The middle figure shows the true and estimated deflection for the identification and validation data set. The bottom figure shows the true and estimated measured force for the identification and validation data set.



(a) Model fitness of a transfer function with two poles and no zeros.



(b) Model fitness of a first order state space system.

Figure 3-5: The measured deflection and force by the UTM compared to the estimated deflection and force using the self-sensing model.

Chapter 4

Modeling

The modeling of the TCPM is discussed in this chapter. A model can be used in the design of a controller for the deflection and the force of the TCPM. Different types of models can be found in the literature for a TCPM such as phenomenological modeling, macroscopic modeling, and black-box modeling. Shafari and Li [2015] propose a phenomenological model describing the microstructural processes during the fabrication of a TCPM. These types of models contain a large number of parameters and include many details. The model of the relation between the actuation strains in molecules caused by the twisting process in the model of Shafari and Li [2015] is not needed to design a proper functioning controller. Therefore only macroscopic modeling and black-box modeling will be discussed here. Afterward, the modeling approach used on the setup (see Section 2-3) is explained.

4-1 Modeling in literature

First, the basic macroscopic modeling of a TCPM found in literature is discussed. Then, the black-box modeling.

4-1-1 Macroscopic modeling

The basic macroscopic model is proposed by Yip and Niemeyer [2015]. It describes the dynamic behavior of a TCPM using mechanical systems such as springs and dampers. The behavior of a TCPM can be described with a series connection of a thermomechanical and a thermoelectric model. This is schematically shown in Figure 4-1 (Yip and Niemeyer [2015]). The input is the power ($P(t)$) and the output can either be chosen as the force generated by the TCPM (F) or the deflection of the TCPM (x). The output depends on the purpose of the TCPM. It can be used in force-control tasks and position-control tasks.

The thermomechanical model describes the relation between strain, force, and temperature. One of the benefits of a TCPM is that its dynamic behavior is less influenced by hysteresis

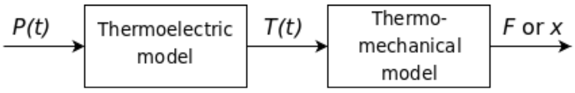


Figure 4-1: Block diagram of a TCPM model with a series connection of a thermomechanical and a thermoelectric model.

than other artificial muscles such as shape memory alloys (Haines et al. [2014]). The hysteresis is mostly visible in the relation between the force and the strain. The obtained force at a certain strain depends on whether the muscle is extending or contracting and how far it has been expanded or contracted before. Yip and Niemeyer [2015] show that the shape of this relation remains constant and that its values shift with temperature. They state that the hysteric behavior can be modeled with a linear spring-damper system. The force can be either increased or decreased by changing the temperature. The thermomechanical model is given by the following equation (Yip and Niemeyer [2015]):

$$F = k(l - l_0) + c\dot{l} + b(T - T_0) \tag{4-1}$$

where l is the loaded length, l_0 the initial length, k the stiffness, c the damping, b a thermal constant, T the temperature of the TCPM, and T_0 the initial temperature of the TCPM.

The thermoelectric model describes the process of Joule heating. Applying a voltage to a metal wire will heat up the wire, because of its electrical resistance. The metal wire is intertwined with the TCPM and increases the temperature of the polymeric fiber. Heat is lost to the environment and must be taken into account. Heating the TCPM is a faster process than cooling the TCPM. The thermoelectric model is given by the following equation (Yip and Niemeyer [2015]):

$$C_{\text{th}} \frac{\mathrm{d}T(t)}{\mathrm{d}t} = P(t) - \lambda(T - T_{\text{amb}}) \tag{4-2}$$

where C_{th} is the thermal mass, $P(t)$ the power applied to the TCPM, λ the absolute thermal conductivity in ambient temperature, and T_{amb} the ambient temperature.

The thermo-electric model is more dominant in the dynamics (Arakawa et al. [2016]). It takes longer to heat and to cool the TCPM than it takes time for the force or deflection to increase or decrease based on the temperature of the TCPM. Other macroscopic models are based on this model. Wu and Tadesse [2017] include the pre-stress of the TCPM in the thermomechanical model and notice that the electrical resistance is commonly used as a constant, but that in reality, it changes over time with temperature. This must be included if a more accurate model is needed. In the self-sensing model, this changing resistance is mentioned but excluded because a constantan wire is used for Joule heating (see Section 3-1-1). Sutton et al. [2016] have adjusted the thermoelectric model by ignoring the effects of radiation. Simeonov et al. [2018] have extended the model to be able to use it for bundled TCPMs. Zhang et al. [2017] have designed a model based on the hysteresis behavior. This model is not discussed further as it is shown that the macroscopic model described in this section can obtain sufficient controlling results. The macroscopic model gives insight into the dynamic behavior of a TCPM, but the exact values of the parameters such as the stiffness and the damping constant are not necessary for designing a functioning controller. Fewer parameters need to be determined when using black-box modeling.

4-1-2 Black-box modeling

A model of a given order is fitted to the input and output data in black-box modeling. It is useful to have an idea of the order of the system to decrease the workload. The thermo-mechanical and thermo-electric model described in the macroscopic modeling can both be described with a first order transfer function. Arakawa et al. [2016] apply this approach in their design of a controller:

$$\begin{aligned} G_{\mathrm{TP}}(s) &= \frac{c_1}{s+c_2} \\ G_{\mathrm{xT}}(s) &= \frac{c_3}{s+c_3} \end{aligned} \tag{4-3}$$

where c_1, c_2, c_3. c_4 are black-box parameters. The complete model of a TCPM has order two because two first order models are connected in series. Four parameters need to be estimated using this black-box model instead of five in the macroscopic model in Section 4-1-1. The input Arakawa et al. [2016] use is the voltage. The voltage needs to be squared and divided by the resistance of the TCPM to obtain power. The output they use is the position. A schematic overview of the model is shown in Figure 4-2.

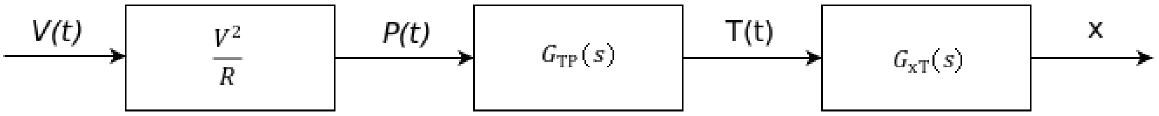


Figure 4-2: Block diagram of the model implemented by Arawaka et. al. Arakawa et al. [2016].

Arakawa et al. [2016] and Yip and Niemeyer [2015] both use second order models. Jafarzadeh et al. [2018] obtain a better fit with a first order discrete-time state-space model than with higher orders. The input they use is the voltage squared and the output is the force. They show that the steady-state relation between the force and the electrical power is close to linear with an error of around 5 %. They also show that the steady-state relation between the voltage squared and the force is close to linear, but with an error of arround 10 %. The accuracy is above 90 % and Jafarzadeh et al. [2018] state that their model is accurate enough to implement a linear controller and that a nonlinear controller is not needed.

4-2 Model estimation

The black-box modeling approach is used to obtain a model for the TCPM. A force-control task is considered and the extension is kept constant. An example of an application is holding a breakable stiff object. The muscle lengths are kept constant and the force needs to be controlled to hold the object without breaking it or letting it fall. The position of the setup is kept constant at 20 mm stretch with the UTM. The zero position of the UTM is when the force of the TCPM is 0.2 N at room temperature (see Section 2-3). The initial length of the

TCPM is approximately 46 mm. The length of the muscle at 0.2 N is approximately 50 mm. This means that the total stretch is approximately 70 − 46 =24 mm, which is an elongation of about 52 %.

First order and second order models are fitted because of the results obtained by Arakawa et al. [2016], Yip and Niemeyer [2015], and Jafarzadeh et al. [2018]. It is uncertain which system order will be a better fit because literature has shown different results. It is expected that a first order system might have a better fit, because the length of the TCPM is kept constant. This will influence the thermo-mechanical model described in Section 4-1-1. The damping term will disappear and the order of the thermo-mechanical model will decrease to zero. The force generated by the TCPM is measured and used as the output signal. The input signal is the duty cycle send to the muscle drive. The input signal is constructed similarly as the input signals used for the self-sensing model, see Section 3-2-1. The same data processing steps have been applied as described in Section 3-2-2. An example of the duty cycle input with the obtained filtered measured force is given in Figure 4-3.

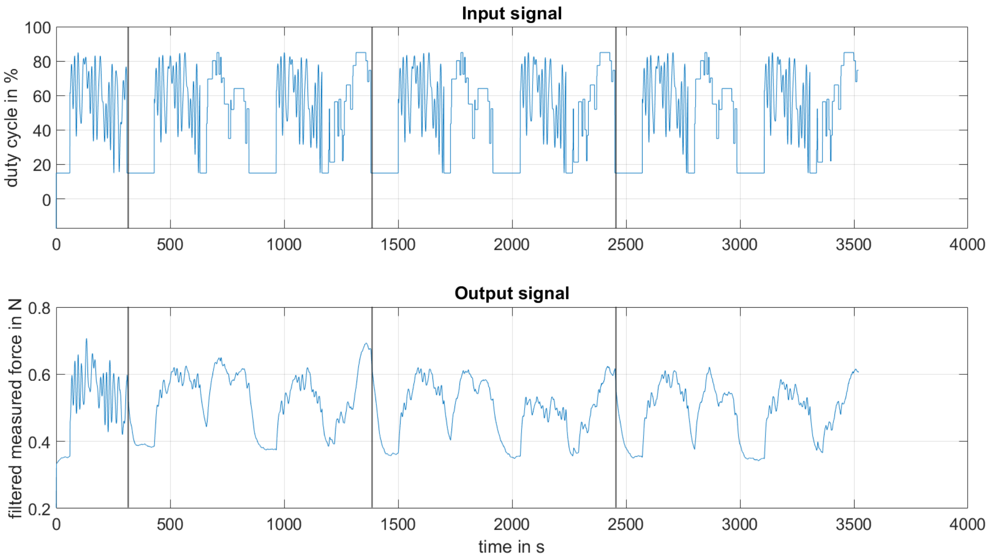


Figure 4-3: The duty cycle used as input signal for model estimation and an example of the obtained filtered measured force.

It can be seen in the figures that the data is separated into four different sets. The first set is the warming up, and will not be used for identification. The other three sets are used for identification and validation. Multiple combinations of the sets have been tried to obtain the model with the best fit. The model estimation is done in MATLAB and uses the toolbox System Identifaction. The data needs to be detrended before the model can be estimated. This is done by using the `detrend`-function. This function removes the best straight-fit line from the data.

Four types of models are used for model estimation. A transfer model is fitted on the detrended data by using the `tfest`-function. A transfer function has the following structure:

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} \tag{4-4}$$

where n is the number of poles and m the number of zeros. These numbers need to be defined in the tfest-function. The function initializes the parameters by using the simplified refined Instrument Variable (SRIV) method described by Young and Jakeman [1980]. The parameters are updated by using a nonlinear least-squared method to minimize the weighted prediction error norm. The nonlinear least-squares method is a combination of line search algorithms. Each method is tried in sequence at each iteration. The first decent direction leading to a reduction in the estimation cost is used. The line search methods are the subspace Gauss-Newton search, the Levenberg-Marquardt least squares search, the adaptive subspace Gauss-Newton search, and the steepest descent least squares search. The algorithms of these methods are given in Appendix B.

Another model that has been fitted is a state-space model. The used function is sses. The order (nx) needs to be defined. The standard form of a state space is:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ke(t) \\ y(t) &= Cx(t) + Du(t) + e(t) \end{aligned} \tag{4-5}$$

where A, B, C, D, and K are state-space matrices. The default value of D is equal to zero. The ssest-function uses a noniterative subspace state-space estimation approach for the initialization of the estimates. It updates the estimates using the prediction error minimization (PEM) approach. The goal is to minimize the cost function with numerical optimization. This is a weighted norm of the prediction error:

$$V_N(G, H) = \sum_{t=1}^{N} e^2(t) \tag{4-6}$$

where $e(t)$ is the error between the measured output and the predicted output. The cost function becomes more accurate for a larger number of data samples (N).

Another model is the estimate output-error polynomial model (oe). This model relates inputs to outputs while also including white noise as an additive output disturbance. The corresponding equation is:

$$y(t) = \frac{B(q)}{F(q)} u(t - n_k) + e(t) \tag{4-7}$$

The oe-function works in a similar manner as the tfest-function. The difference is that in the oe-function the model is specified with polynomial degrees instead of the numbers of poles and zeros. The polynomial degrees that need to be specified are the order of the polynomial $B(q)$ (n_b), the order of the polynomial $F(q)$ (n_f), and the input delay (n_k). The tfest-function is faster and more accurate in continous-time estimation according to MATLAB.

The last model is the autoregressive with extra input (ARX) model. This model is also known as the autoregressive with exogenous variables model. The structure of the model is given by:

$$y(t) + a_1 y(t-1) + \ldots + a_{na} y(t - na) = b_1 u(t - nk) + \ldots + b_{nb} u(t - nb - nk + 1) + e(t) \tag{4-8}$$

where na is the number of poles, nb is the number of zeros, nk is the dead time and $e(t)$ is the white-noise disturbance value. The dead time is the number of input samples that occur before the input affects the output. The `arx`-function minimizes the one step ahead prediction error between the measured en predicted outputs. The parameters vector (θ) is estimated with the following equation:

$$\theta = (J^T J)^{-1} J^T y \tag{4-9}$$

where J is the regressor matrix.

The obtained models are also updated with other data sets to see if the accuracy of the models increases. This is done by using the `pem`-function. This function updates the parameters of the initial model to fit the given data. It uses prediction-error minimization (described at the state-space model) to update the parameters.

4-3 Results

First order and second order models have been tried to find the model with the highest fit. This is based on what is found in the literature. The Normalized Root Mean Square Error (NRMSE) is used to analyze the goodness of the fit. First, a closer look is taken into the transfer function models. First and second order models are estimated with each data set described in Section 4-2. The model is then validated against all three data sets by using the `compare`-function. This function simulates the response of the model and compares it to the measurement data. It calculates the NRMSE and plots all the data in a figure. These figures are shown in Figure 4-4. The titles of the figures describe which data set is used for validation. In the legend, the transfer functions are labeled. The first number describes the data set used for identification. The number after the underscore describes the order of the transfer function. No zeros are used.

The first and second order models have similar NRMSE values, but the first order models have most of the time a slightly better fit. It was expected that the first order models would have a significantly higher NRMSE value because the length is kept constant. This expectation is also based on the results Jafarzadeh et al. [2018] obtained. The models are updated using the `pem`-function and the two data sets not used for identification. The resulting figures are shown in Figure 4-5. The performance of the models mostly slightly increase with an order of 0-1 %. Again, the first order models have a slightly better fit than the second order models.

This process is repeated for multiple muscles. Figures from another muscle is shown in Appendix A-2 and multiple muscles in Appendix A-3. The trend that the first order models have a slightly better fit is visible in all muscles. Appendix A-3 include higher order models.

First and second order state-space models have been fitted to the data as well. The resulting figures are shown in Figure 4-6. The results obtained after updating the models is shown in Figure 4-7. The obtained models are less reliable in comparison to the transfer function models. Some of the second order models do not describe the behavior of the TCPM at all. The first order models score on average again a higher NRMSE. Figures obtained by using the data of another muscle are shown in Appendix A-4. In Appendix A-5 other orders have

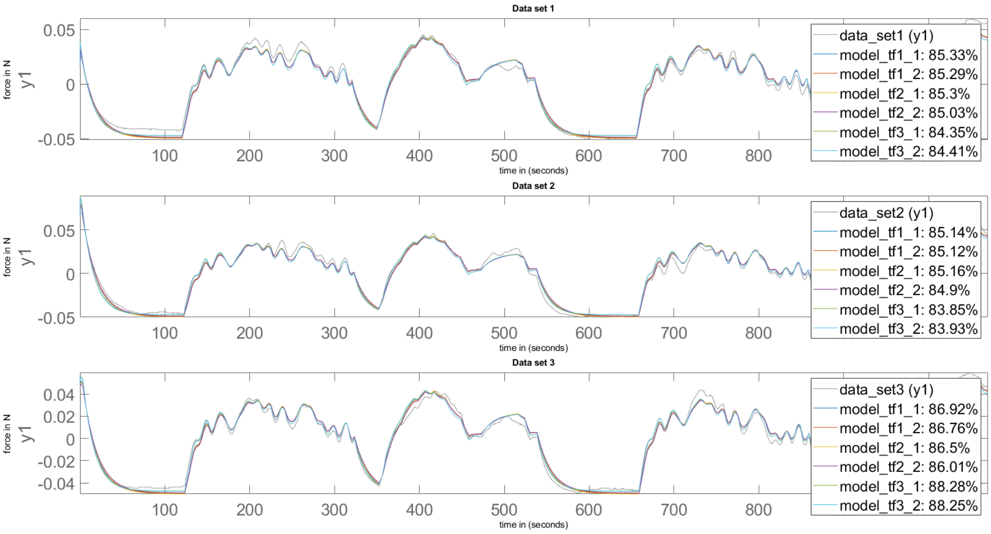


Figure 4-4: Performance of the estimated transfer functions. The titles of the figures describe which data set is used for validation. In the legend, the transfer functions are labeled. The first number describes the data set used for identification. The number after the underscore describes the order of the transfer function. No zeros are used.

been tried next to first and second order models. The same trend of less reliable second order state-space models can be seen.

Jafarzadeh et al. [2018] obtained at least 90 % accuracy with the first order model. The fit is less accurate for the produced TCPMs. It is chosen to continue with first order models due to the decrease in complexity. This is based on the similar NRMSE values of first and second order models and the results obtained by Jafarzadeh et al. [2018]. First order arx and oe models are also estimated to see which type of model has the best fit. The same division of the data set is used. The oe, arx, tf, and ss models with the best fit are compared to each other. The results of this are shown in Figure 4-8. The mean and standard deviation for the models are given in Table 4-1.

Table 4-1: The mean and standard deviation of the NRMSE of the estimated models.

Model	mean (μ)	standard deviation (σ)
tf	79.26	3.66
ss	79.30	3.26
oe	79.15	2.95
arx	79.34	3.27

Next, step responses are included in the data sets. This is because first order models are used. The data of the step responses is the fourth data set used in identification and validation. An example of figures obtained with the step responses is shown in Figure 4-9. Comparing the mean and standard deviations of multiple models the transfer function and the oe-models have the best fit. Therefore, only those two types of models are used in further model estimation. The best transfer function has a fit of 73.37±4.15 %. The best oe-model has a

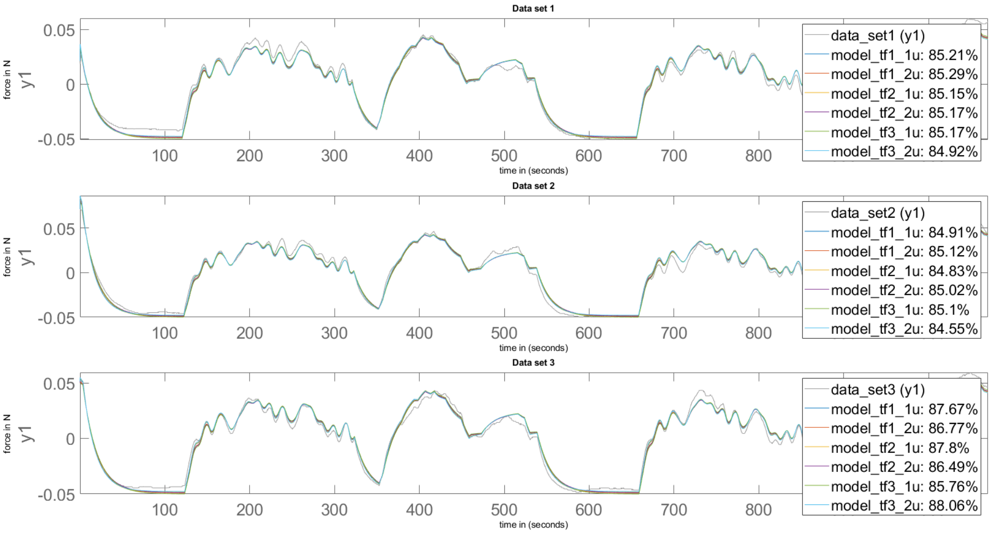


Figure 4-5: Performance of the updated transfer functions. The titles of the figures describe which data set is used for validation. In the legend, the transfer functions are labeled. The first number describes the data set used for identification. The number after the underscore describes the order of the transfer function. No zeros are used.

fit of 73.36±4.10 %. These NRMSE values are lower than previously obtained because the second data set shows different behaviour than in other runs. The poles and time constants of the models are shown in Table 4-2.

Table 4-2: The poles, damping, frequency ant time constant for the two best models.

Model	Pole	Time constant (s)
tf	-0.0432	23.1
oe	-0.0437	22.9

Next, two elongation lengths are used. The first is the 52 % described in Section 4-2. The other is of approximately 20 %. This means that the position of the UTM is set to 5 mm instead of 20 mm. This is done to check whether a model obtained with a certain elongation can be used on different elongations. It is expected that the NRMSEs value will decrease because first order models are identified. The difference in length leads to a higher order of the differential equation of the thermo-mechanical model described in Section 4-1-1. The transfer function and the oe-model with the highest fit for the 52 % elongation is validated on the data of the 20 % elongation and vice versa. The results are shown in Figure 4-10. The models with number 1 use data from the 20 % elongation and the models with number 4 use data from the 52 % elongation. The fit significantly decreases when the elongation is varied. This means that if the muscle is used at a different elongation, the model is less accurate and the controller designed for it will function less efficiently.

The poles and the time constants for the models are shown in Table 4-3. The larger elongation has faster poles and a lower time constant than the shorter elongation. The cooling down process is faster when the TCPM is stretched out more because the heat from the coils has

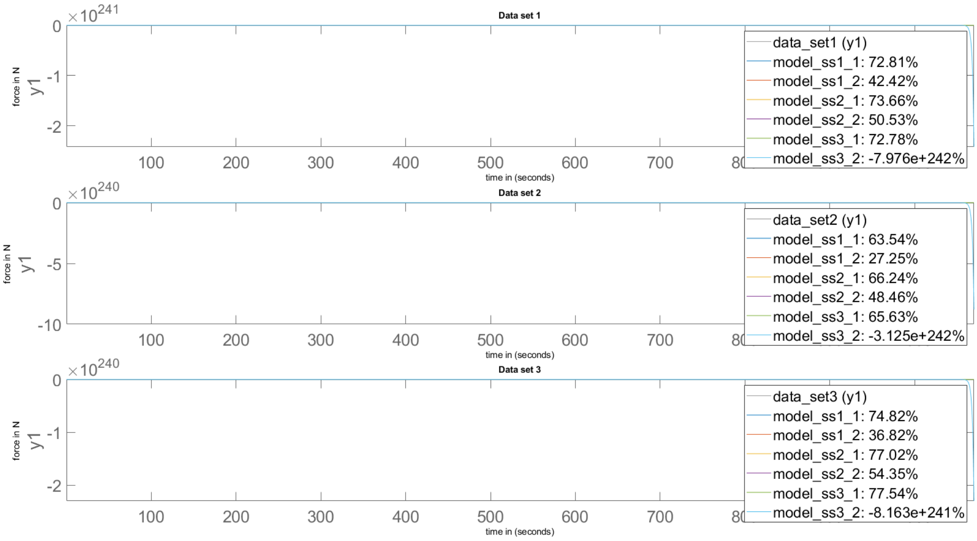


Figure 4-6: Performance of the estimated state-space models. The titles of the figures describe which data set is used for validation. In the legend, the state-space models are labeled. The number describes the data set used for identification.

less influence on each other.

Table 4-3: The poles, damping, frequency ant time constant for the two best models.

Model	Pole	Time constant (s)
tf1u	-0.0515	19.4
oe1u	-0.0515	19.4
tf4u	-0.0995	10.0
oe4u	-0.0994	10.1

First order transfer functions and oe-models have the best fit for the produced TCPM. The performances of the transfer functions and oe-models are comparable. It is chosen to continue with the transfer functions, because of its implementation in Simulink. Last, models of two different TCPM are validated with the data of the other TCPM. It is useful to know whether for each produced TCPM the model estimation process needs to be repeated or the same model can be used for different TCPM. The used elongation is 52 %. The accuracy of the model significantly decreases when used on another TCPM. This is shown in Figure 4-11. This means that model estimation has to be repeated for each produced muscle.

In conclusion, the chosen model is a first order transfer function. A different model has to be estimated for each elongation and each TCPM.

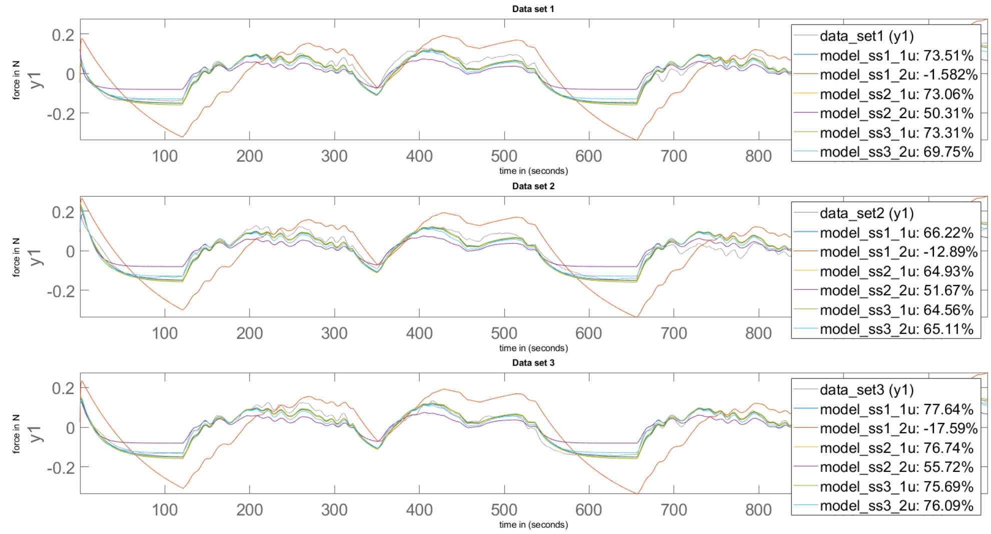


Figure 4-7: Performance of the updated state-space models. The titles of the figures describe which data set is used for validation. In the legend, the state-space models are labeled. The number describes the data set used for identification.

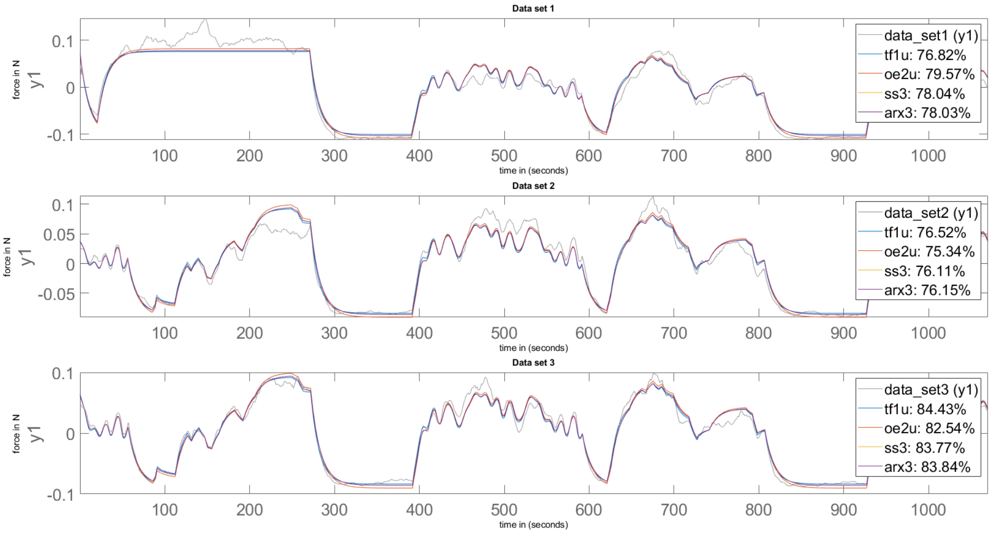


Figure 4-8: Performance of the estimated models. The titles of the figures describe which data set is used for validation. The u stands for the updated version.

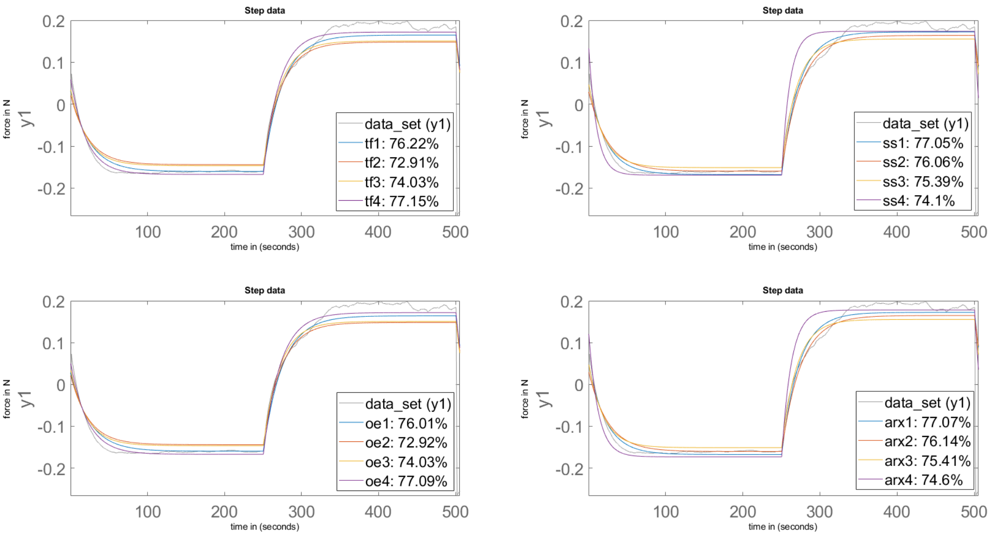


Figure 4-9: Performance of the estimated models on the step data. In the legends, the models are labeled. The number describes the data set used for identification.

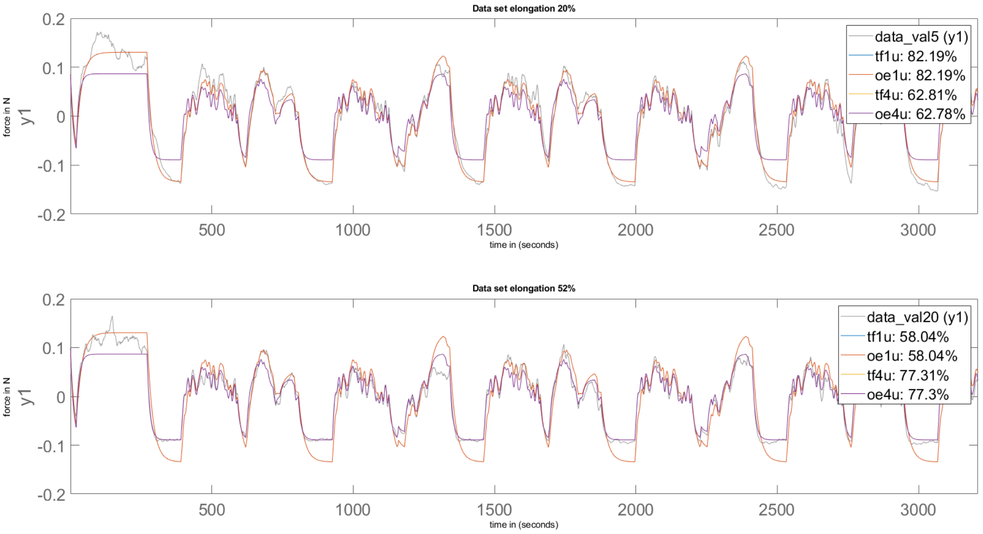


Figure 4-10: Performance of the estimated models on different elongations.

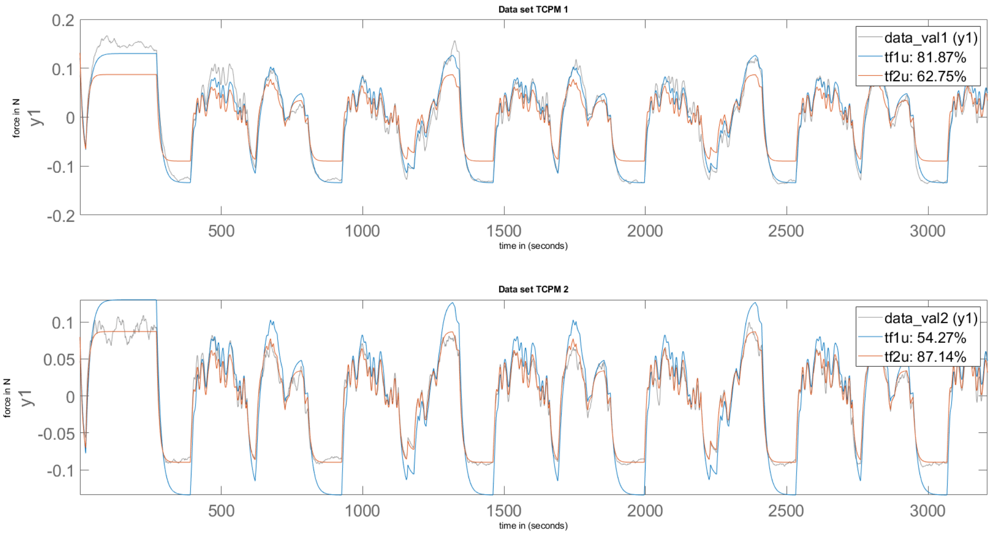


Figure 4-11: Performance of the estimated models on different TCPM.

Chapter 5

Control

The used model is a transfer function of first order. The force of the TCPM needs to be controlled. The UTM keeps the deflection constant. This is called a force-control task. An example of an application is holding a breakable stiff object. The muscle lengths are kept constant and the force needs to be controlled to hold the object without breaking it or letting it fall. The correct force needs to be applied. The self-sensing model is not included in control as explained in Section 3-2-4. First, a P-controller is designed. Then, a PI-controller.

5-1 P-controller

A high gain P-controller is designed for the TCPM. The controller is tested on the setup. First, several step references are applied. The results are shown in Figure 5-1. It can be seen that the obtained force oscillates around the reference value. The heating and the cooling down of the TCPM take approximately the same time. This is because of the cooling fan. Without the fan, the cooling down would take more time. The input is quite on-off, because of the high K_p-value.

The controller is also tested on a sinusoidal reference. The results are shown in Figure 5-2. First, some step references are applied. It can be seen that when the reference goes from 0.5 N to 0.3 N the TCPM does not have enough time to cool down. The time constants of the model are slow, which can be impractical when the TCPM is implemented in for instance a robot. The sinusoidal reference is followed with high accuracy. The filtered force still oscillates around the reference leading to an oscillation in the sinusoidal.

The oscillations show that the system is actually of a higher order, thus order 2. The oscillations also occurred in the results of the P-controller obtained by Jafarzadeh et al. [2018], but less frequent. An I-action is added to decrease the min-max input.

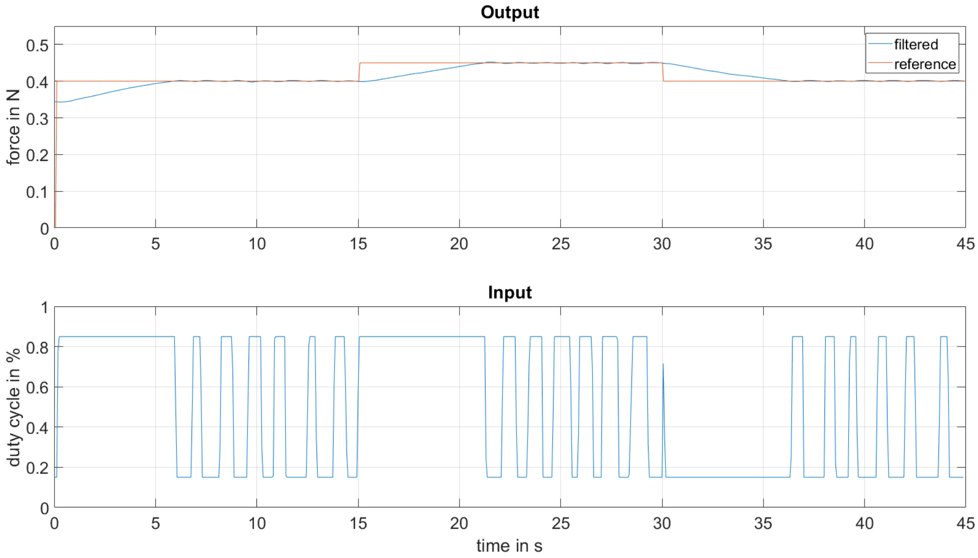


Figure 5-1: Performance of the P-controller on step references.

5-2 PI-controller

The PI-controller is designed via pole-placement. The controller values can be determined with:

$$K_p = \frac{-p_1 - p_2 + p_{\text{tf}}}{K} \quad K_i = \frac{-p_1 * -p_2}{K} \tag{5-1}$$

where p_1 and p_2 are the desired poles, p_{tf} is the pole of the model, and K is the gain of the model.

Several pole locations have been tried. To explain the response, three pole sets are discussed. First, two poles on -100 are applied. The obtained controller values are $K_p = 1.06e4$ and $K_i = 5.29e5$. The simulated response is shown in Figure 5-3. There is a large overshoot and the heating and cooling down processes are too fast for the real system. The input necessary is given in Figure 5-4. In the beginning, there is still a lot of on-off input.

Next, the desired poles are placed at -10. The obtained controller values are $K_p = 1.06e3$ and $K_i = 5.29e3$. The simulated response is shown in Figure 5-5. The oscillations are damped faster, but still, the speed of increasing and decreasing the force is not achievable on the setup. The input needed is shown in Figure 5-6. The input is still on and off.

Last, the desired poles are placed on -1. The obtained controller values are $K_p = 103$ and $K_i = 52.9$. These values are more realistic to implement. The simulated response is shown in Figure 5-7. The oscillations have decreased further. The cooling down process shown in the step response is still not achievable on the practical implementation. The input signal is given in Figure 5-8. The on-off behavior has decreased. The results of this simulation is the most promising. Therefore, it is chosen to implement this controller on the experimental setup.

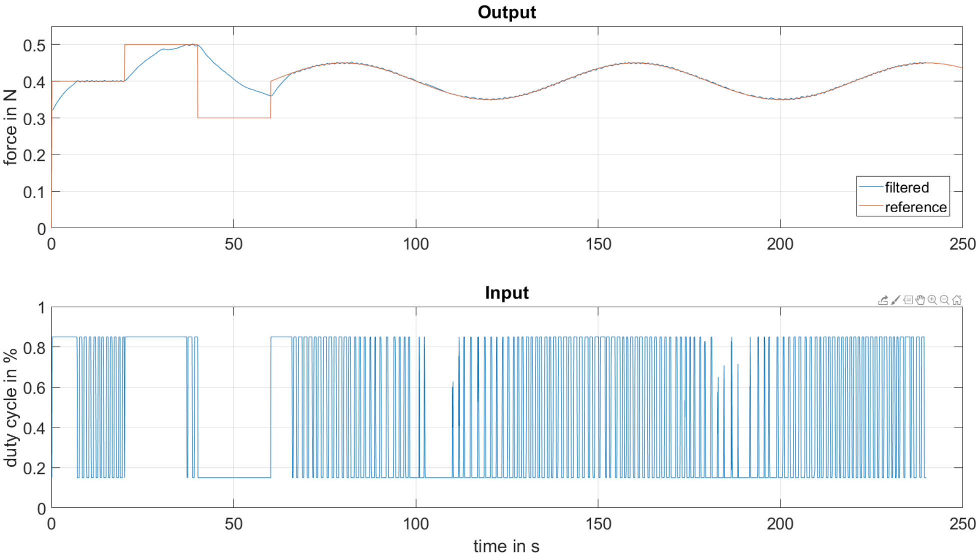
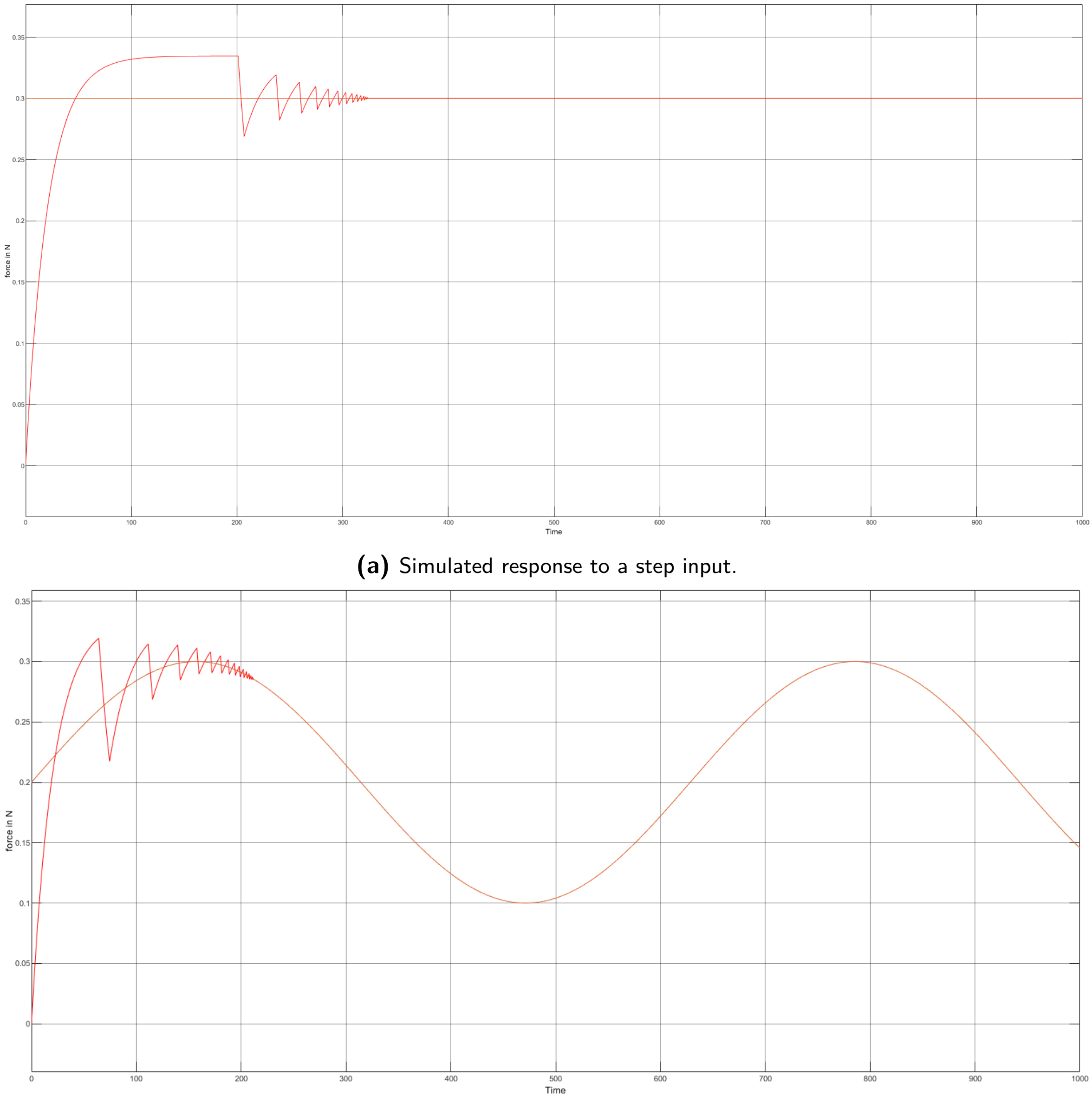


Figure 5-2: Performance of the P-controller on a sinusoidal reference.

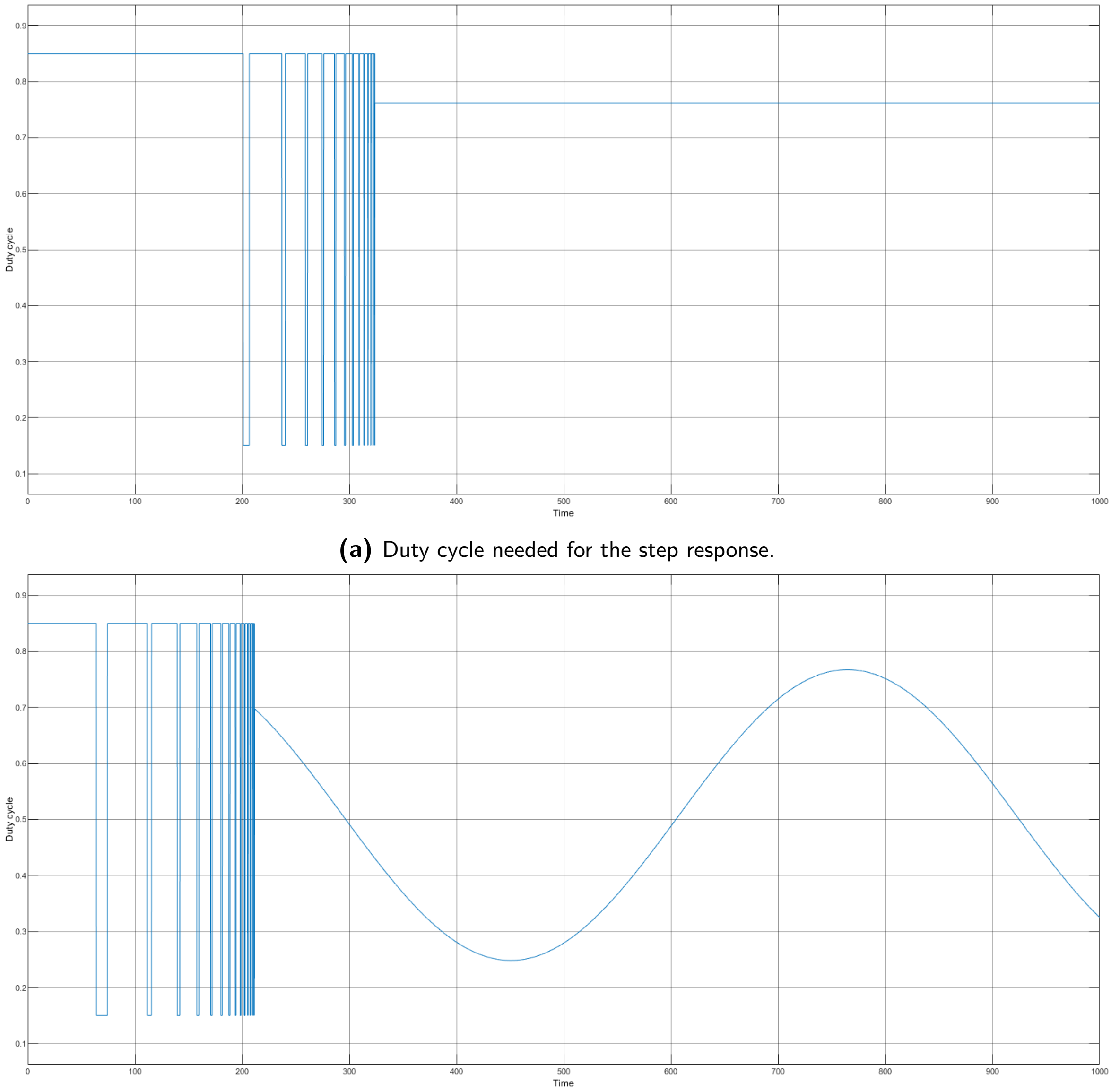
The PI-controller used on the setup has the values $K_p = 103$ and $K_i = 53$. The performance of the PI-controller on the setup is given in Figure 5-9. It can be seen that the force still oscillates around the reference value. The frequency of this oscillation has decreased compared to the P-controller. The input needed is also less on-off than with the P-controller. A D-action is needed to dampen the oscillations. This can be done in further research. The I-action can also be extended to an anti-windup.



(a) Simulated response to a step input.

(b) Simulated response to a sinusoidal input.

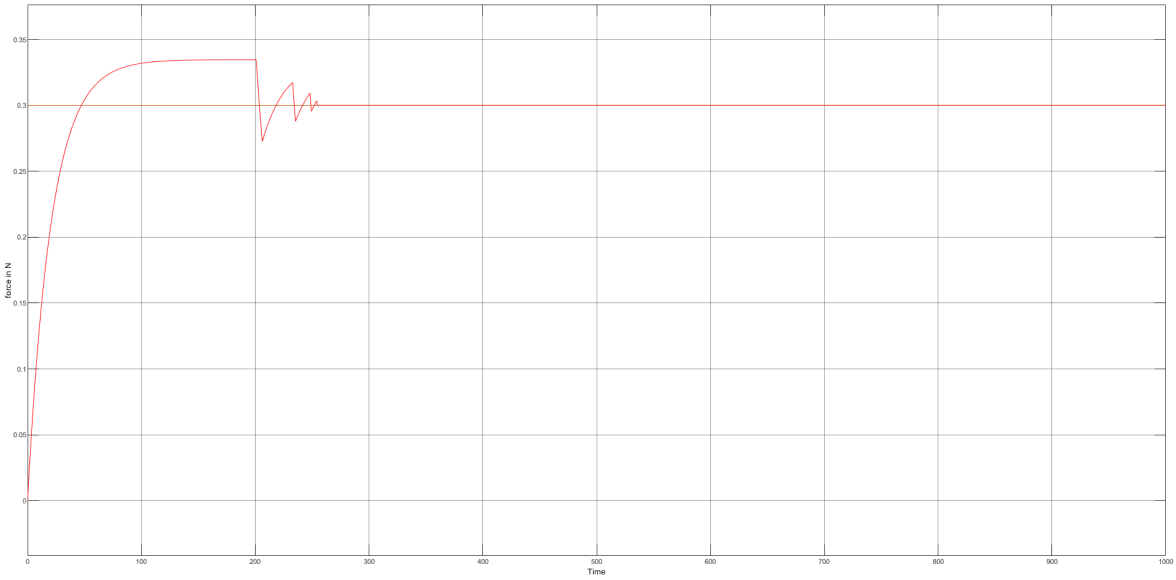
Figure 5-3: Simulated responses for the system with two poles at -100.



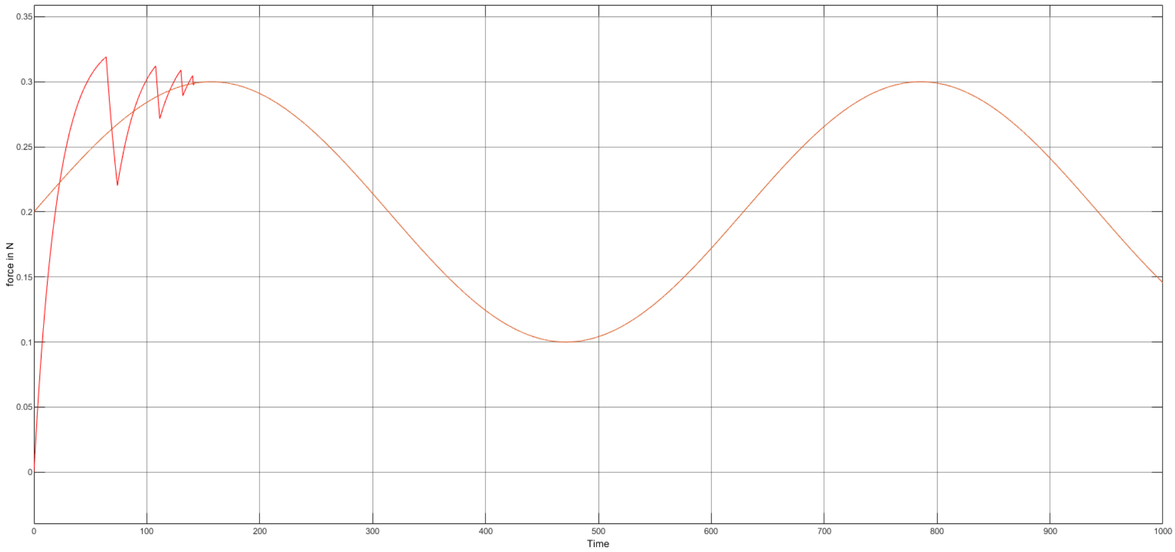
(a) Duty cycle needed for the step response.

(b) Duty cycle needed for the sinusoidal reference.

Figure 5-4: Simulation of the duty cycle needed for the system with two poles at -100.

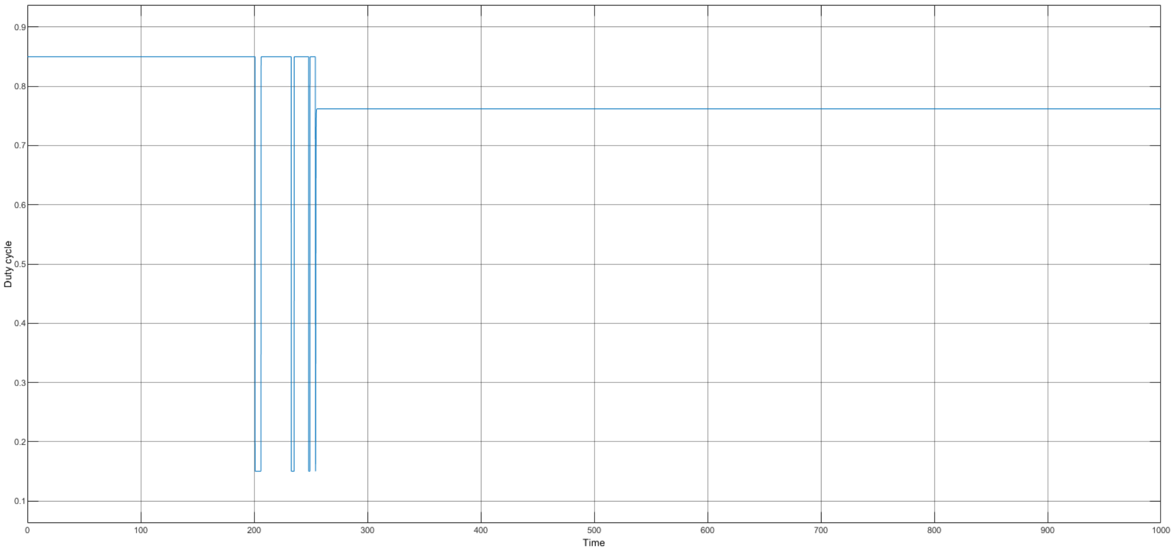


(a) Simulated response to a step input.

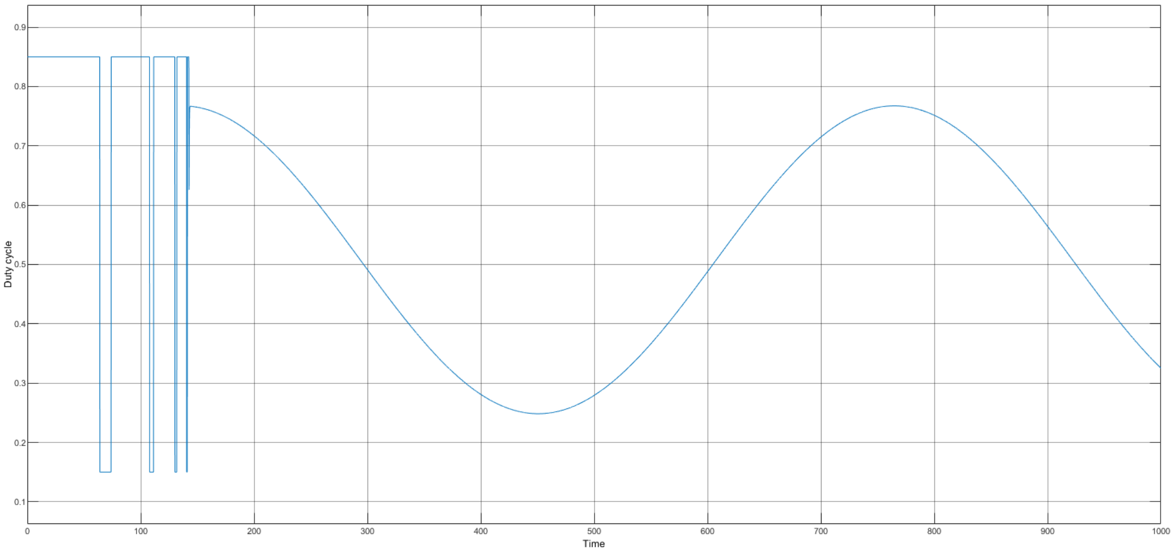


(b) Simulated response to a sinusoidal input.

Figure 5-5: Simulated responses for the system with two poles at -10.

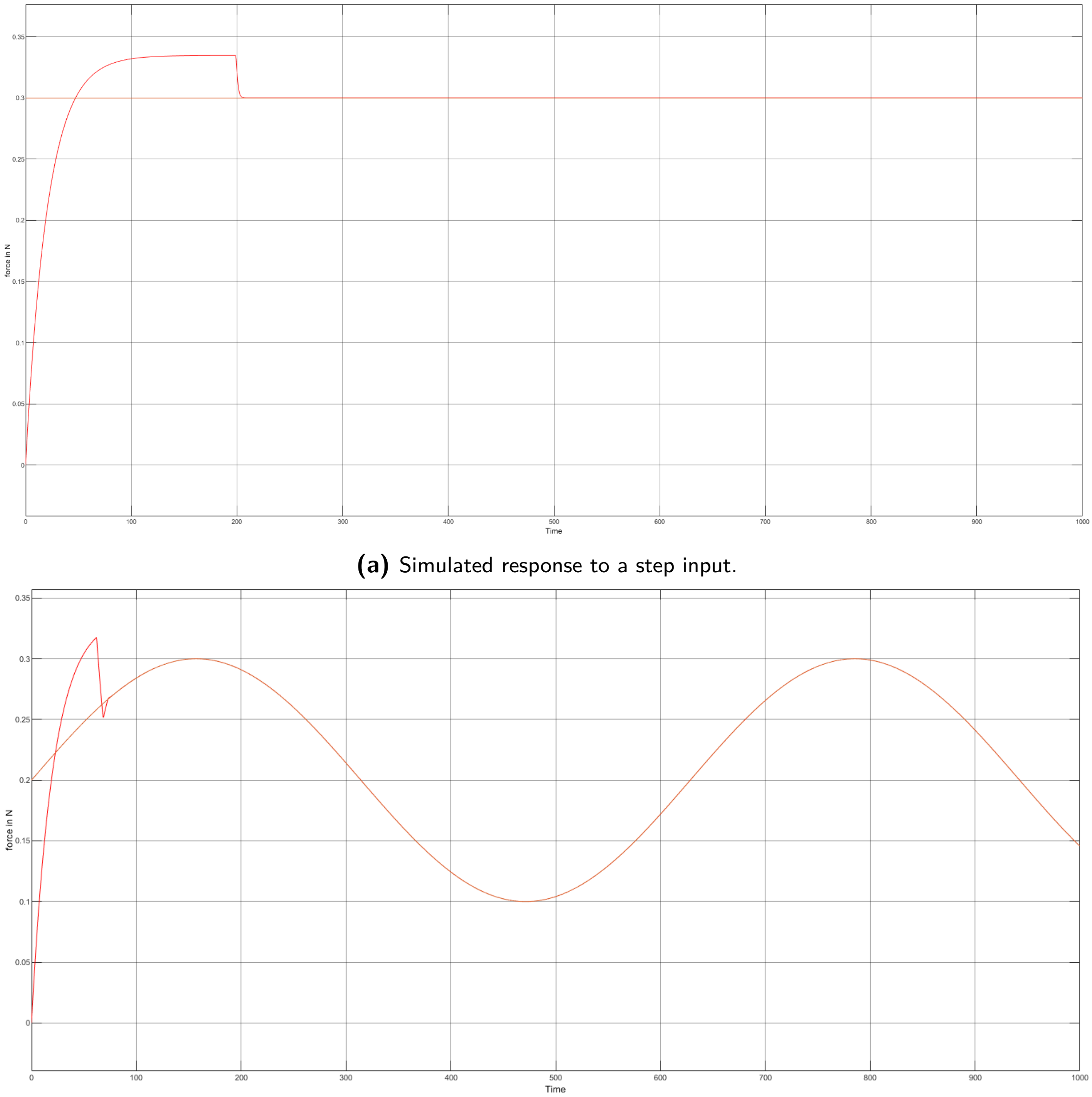


(a) Duty cycle needed for the step response.



(b) Duty cycle needed for the sinusoidal reference.

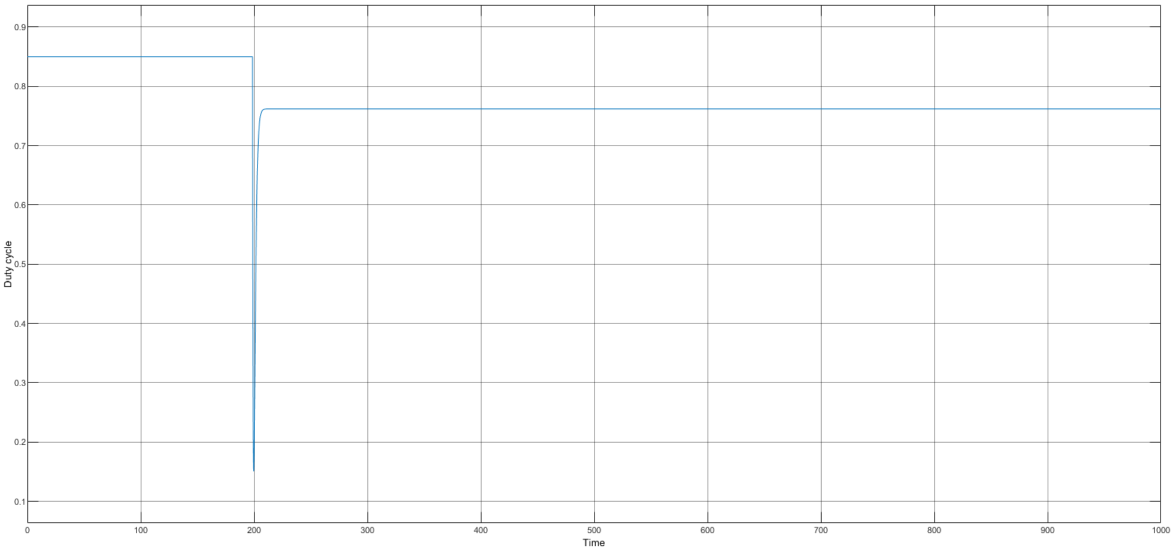
Figure 5-6: Simulation of the duty cycle needed for the system with two poles at -10.



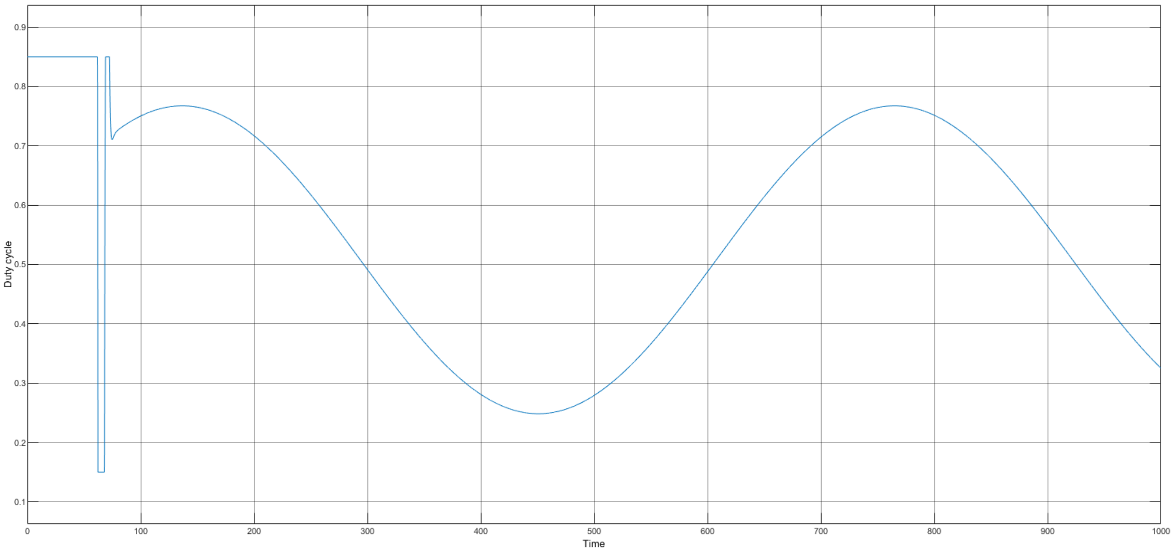
(a) Simulated response to a step input.

(b) Simulated response to a sinusoidal input.

Figure 5-7: Simulated responses for the system with two poles at -1.



(a) Duty cycle needed for the step response.



(b) Duty cycle needed for the sinusoidal reference.

Figure 5-8: Simulation of the duty cycle needed for the system with two poles at -1.

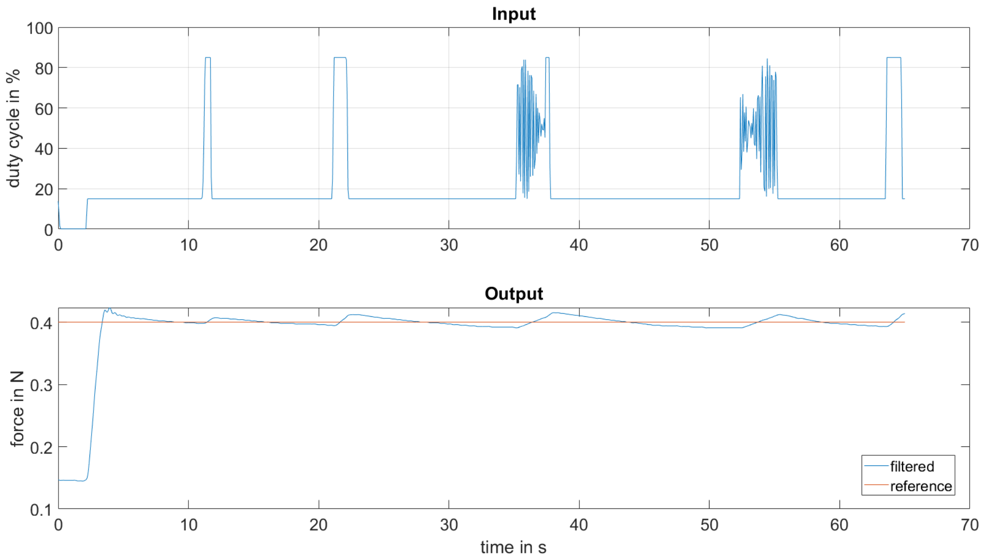


Figure 5-9: Performance of the PI-controller.

Chapter 6

Conclusions and further research

The goal of this thesis is to implement a controller to control the force of a TCPM that works with the self-sensing model. It is shown in Chapter 3-2-4 that the self-sensing model is not accurate enough to be used for controller design. Van der Weijde et al. [2019] obtains more accurate results with the self-sensing model. An explanation of this can be that the production process of the TCPMs is not automated. This means there is a higher chance of inconsistencies between TCPMs. The TCPMs need to be twisted at a constant velocity. This is difficult to achieve with a drill controlled by a human. One person can be better at this job than another. Further research can be done into whether this plays a role in the accuracy of the self-sensing model.

A first order black-box model is fitted to the input and output data of the TCPM. The input is the duty cycle and the output is the force. It is shown in Chapter 4-3 that the first order models achieve the best fit. This follows results obtained by Jafarzadeh et al. [2018]. The accuracy is less. The cause can be that Jafarzadeh et al. [2018] use a silver plated nylon wire for Joule heating instead of a metal resistance wire. They also use still air instead of moving air. Moving air decreases the cooling time. Another difference is that Jafarzadeh et al. [2018] use a state-space model instead of a transfer function. Other literature has shown second order models, but the order is decreased when keeping the stretch of the TCPM constant. The fitted first order transfer function is accurate enough for controller design.

First, a P controller has been implemented. The measured force oscillates around the reference force indicating that the actual model is of second order. Jafarzadeh et al. [2018] obtained similar results. The input shifts from max to min and vice versa. An integral action is added to reduce this behavior. The force follows step signal and sinusoidal references. It still oscillates around the reference. To reduce this a D-action needs to be added.

The parameters of the model and therefore controller is different for each TCPM. This is impractical in large-scale applications, such as assistive wrist orthosis and humanoid robots. Further research can be done in designing model-free controllers and methods such as neural networks.

Appendix A

Figures

A-1 Fit TCPM2

The fit of the self-sensing model for TCPM 2.

A-2 Performance of first and second order transfer function models

Figures showing the performance of first order and second order transfer function estimations are shown in this section. The results are similar with what is described in Section 4-3.

A-3 Performance of higher order transfer function models

Figures showing higher order transfer functions. The number behind the transfer function indicates the order. Data set 1 is used for estimation.It can be seen that first, second and third order models have the best fit. Because of literature, third order models are not included further.

A-4 Performance of second order state-space models

A-5 Performance of higher order state-space models

Figures showing higher order transfer functions. The number behind the transfer function indicates the order. Data set 1 is used for estimation.It can be seen that the fit of higher order models is not consistent. Based on literature, only first and second order models are further estimated.

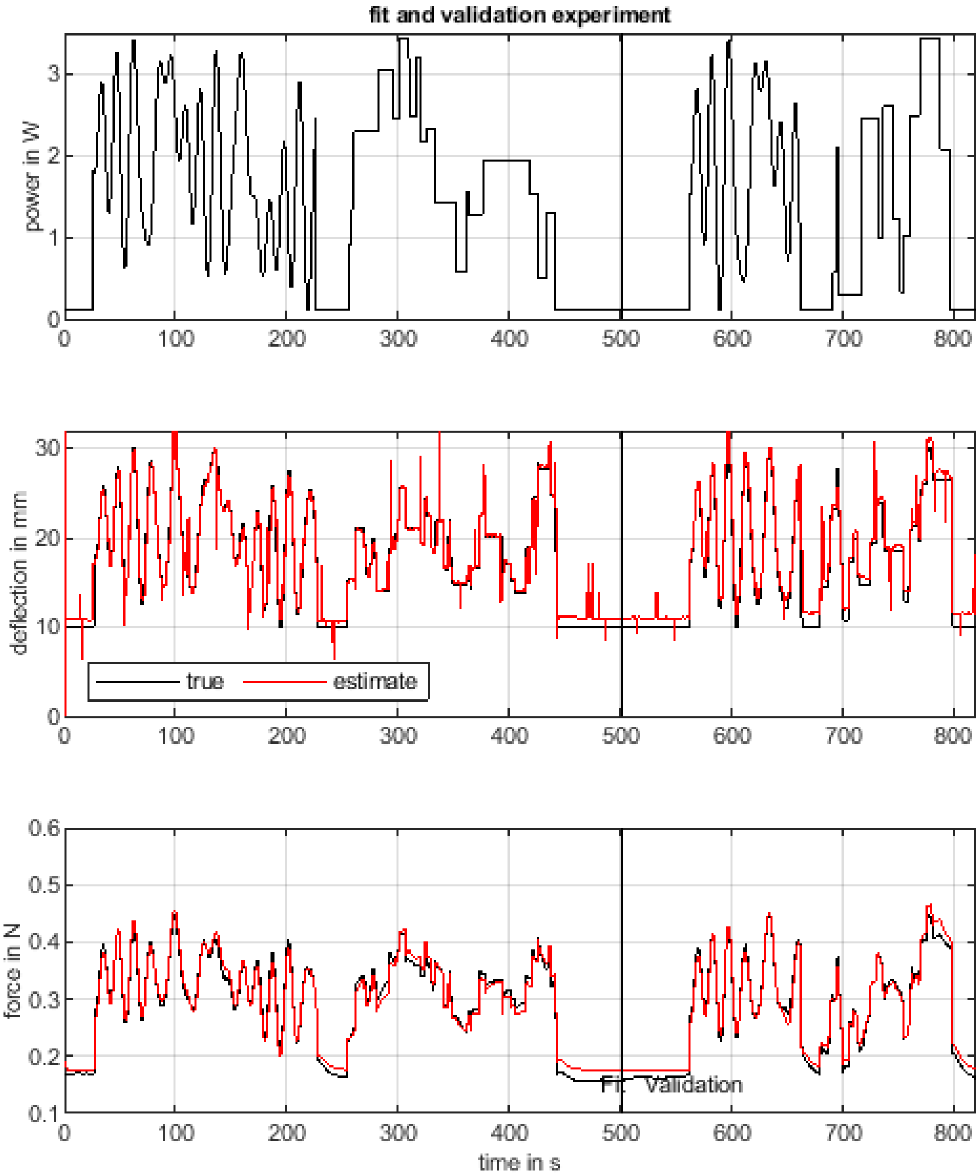


Figure A-1: The top figure shows the applied power over time. The middle figure shows the true and estimated deflection for the identification and validation data set. The bottom figure shows the true and estimated measured force for the identification and validation data set.

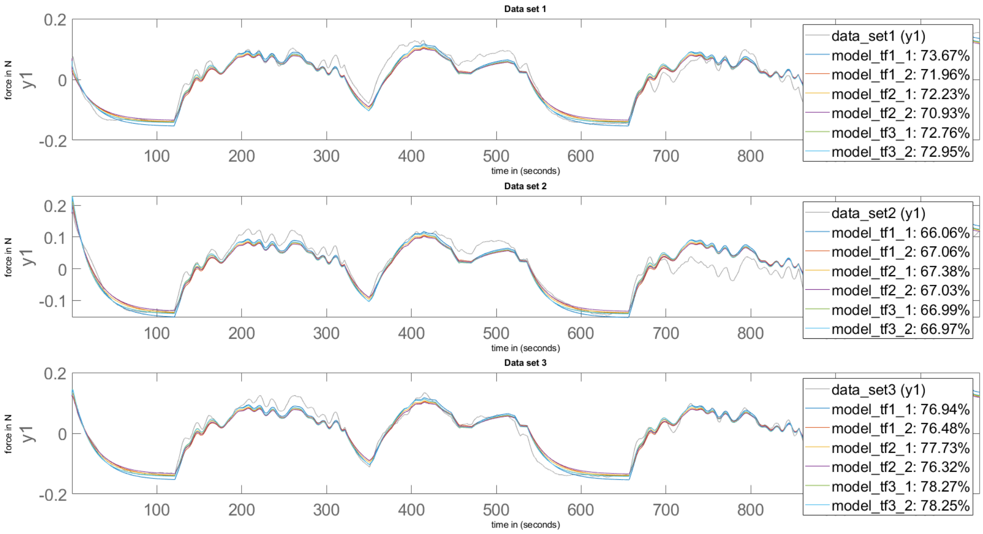


Figure A-2: Performance of the estimated transfer functions. The titles of the figures describes which data set is used for validation. In the legend, the transfer functions are labeled. The first number describes the data set used for identification. The number after the underscore describes the order of the transfer function. No zeros are used.

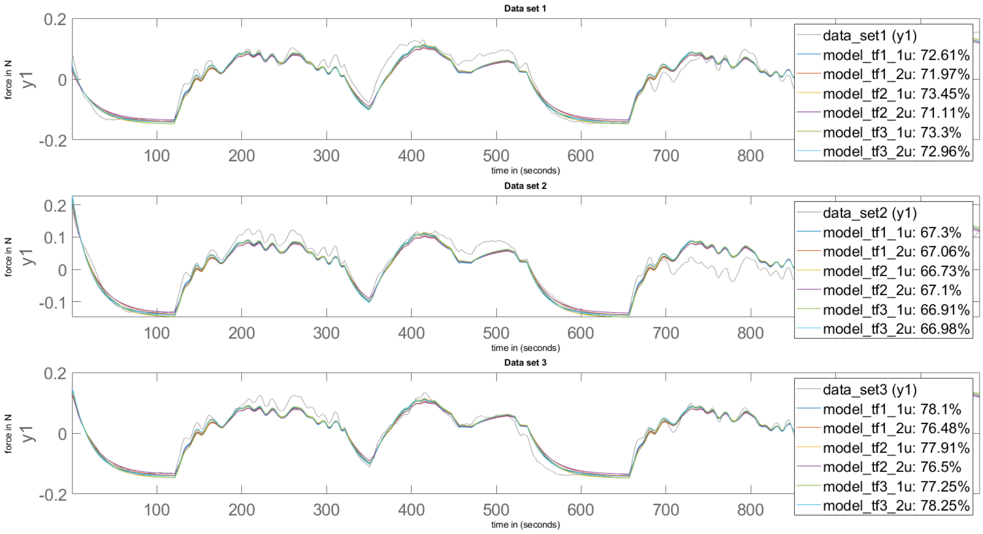


Figure A-3: Performance of the updated transfer functions. The titles of the figures describe which data set is used for validation. In the legend, the transfer functions are labeled. The first number describes the data set used for identification. The number after the underscore describes the order of the transfer function. No zeros are used.

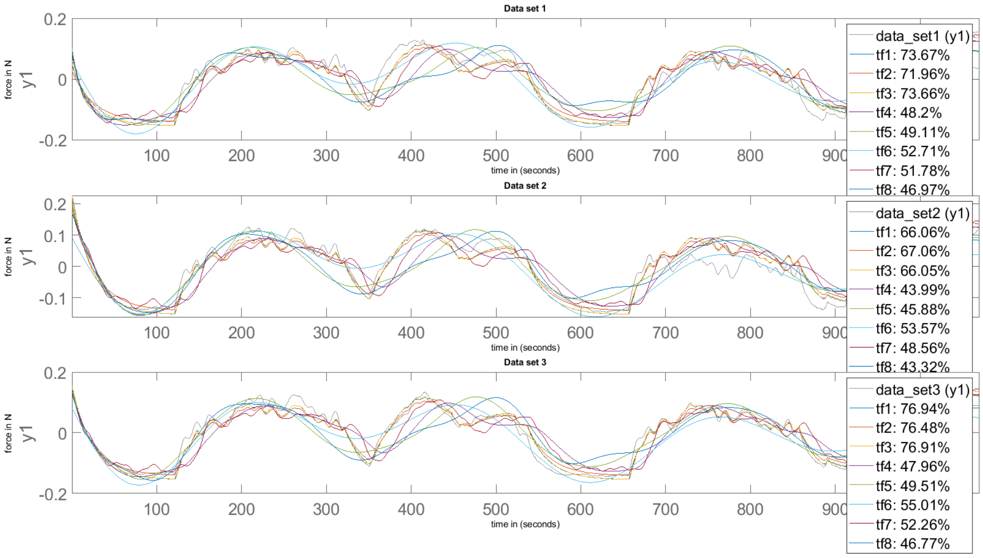


Figure A-4: Performance of transfer functions with different orders. No zeros are used.

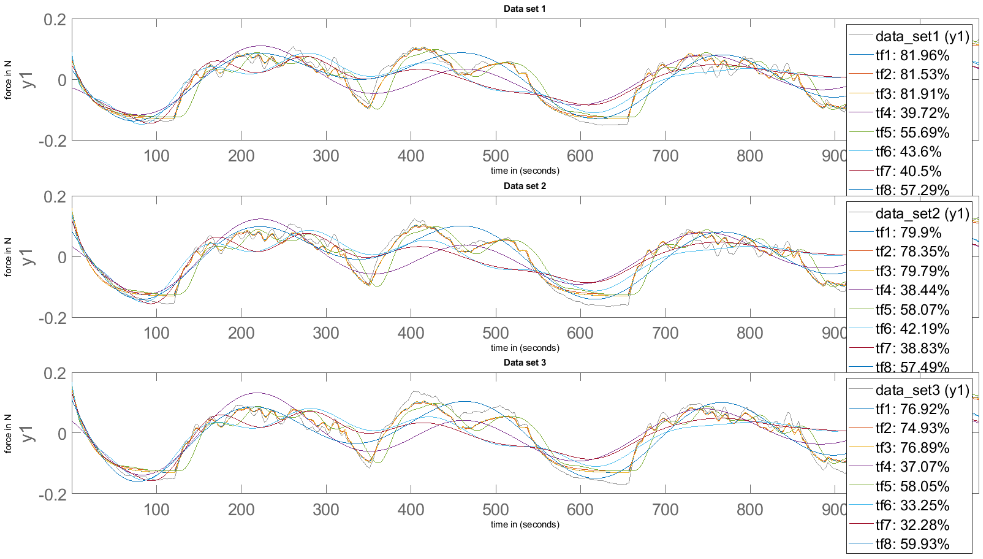


Figure A-5: Performance of transfer functions with different orders. No zeros are used.

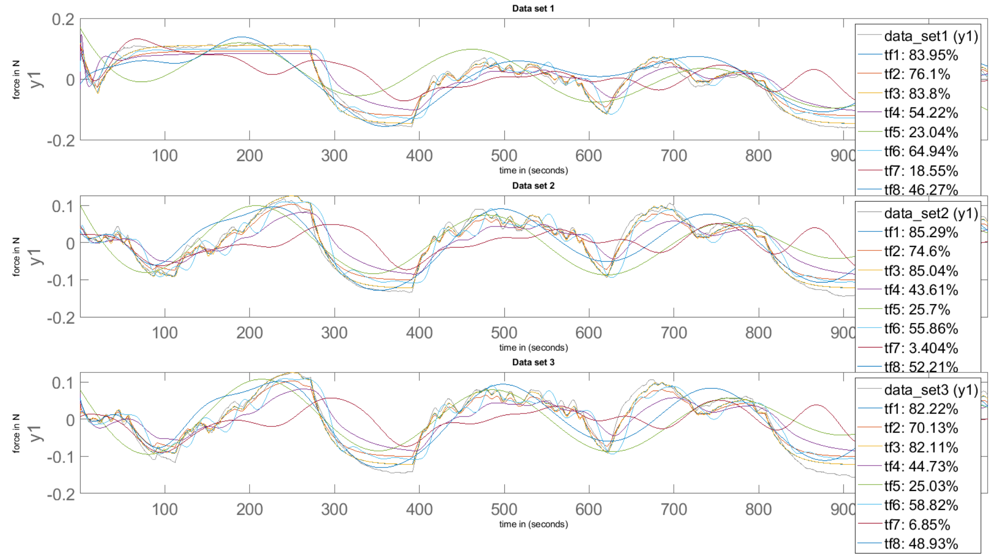


Figure A-6: Performance of transfer functions with different orders. No zeros are used.

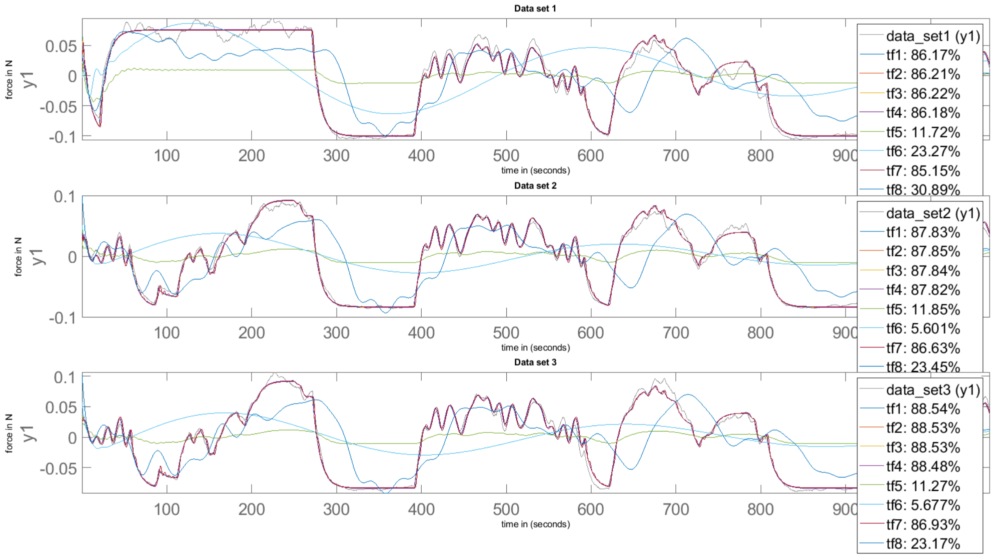


Figure A-7: Performance of transfer functions with different orders. No zeros are used.

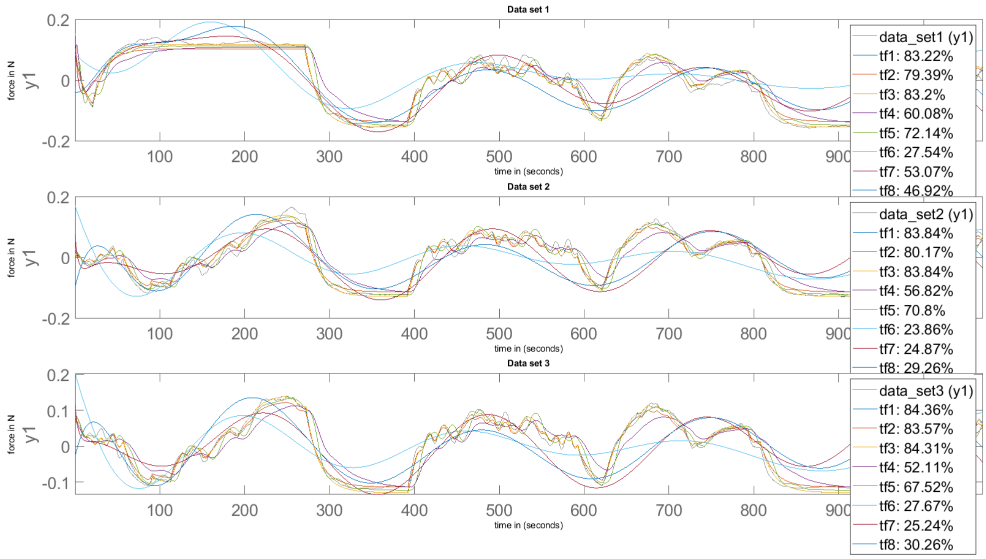


Figure A-8: Performance of transfer functions with different orders. No zeros are used.

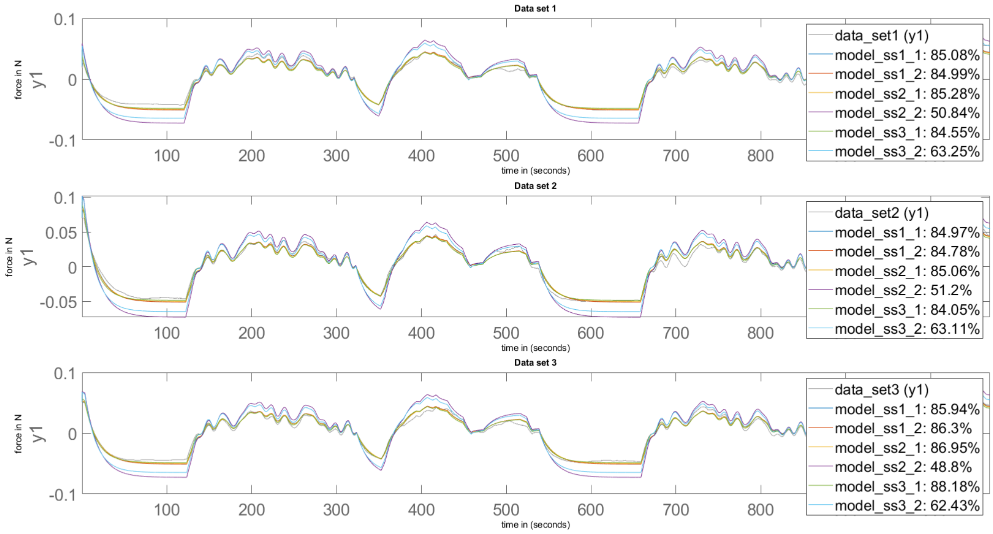


Figure A-9: Performance of the estimated state space models. The titles of the figures describe which data set is used for validation. In the legend, the state-space models are labeled. The number describes the data set used for identification.

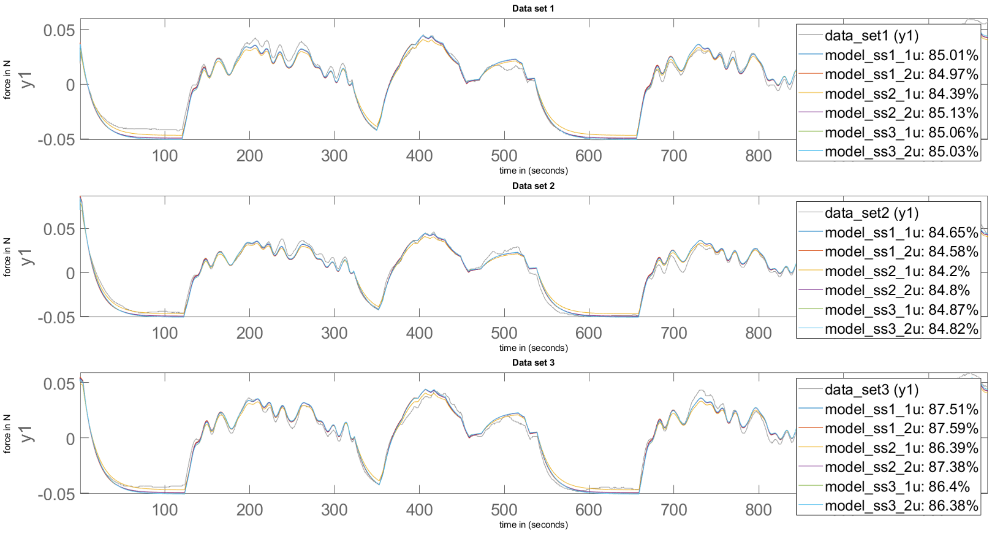


Figure A-10: Performance of the updated state space models. The titles of the figures describe which data set is used for validation. In the legend, the state-space models are labeled. The number describes the data set used for identification.

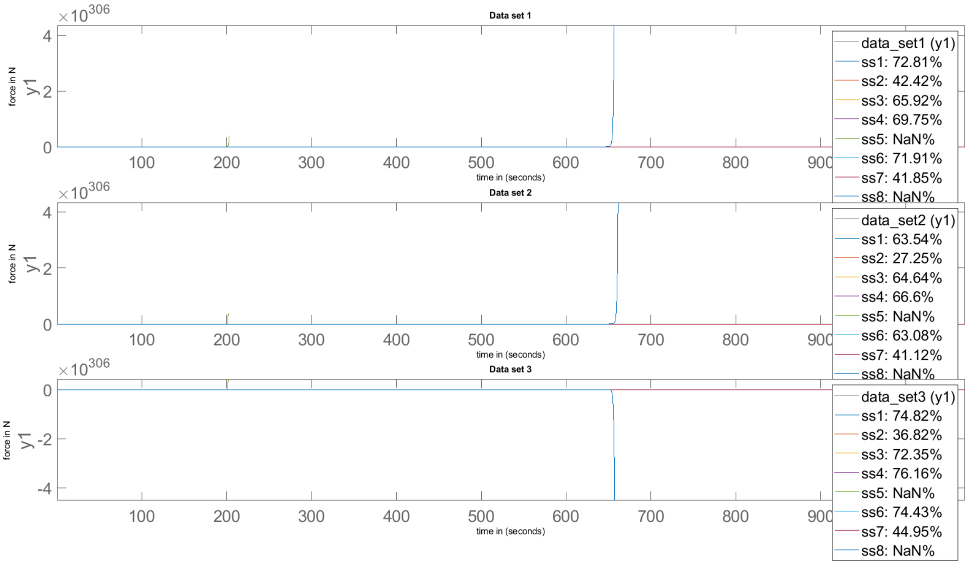


Figure A-11: Performance of state space models with different orders.

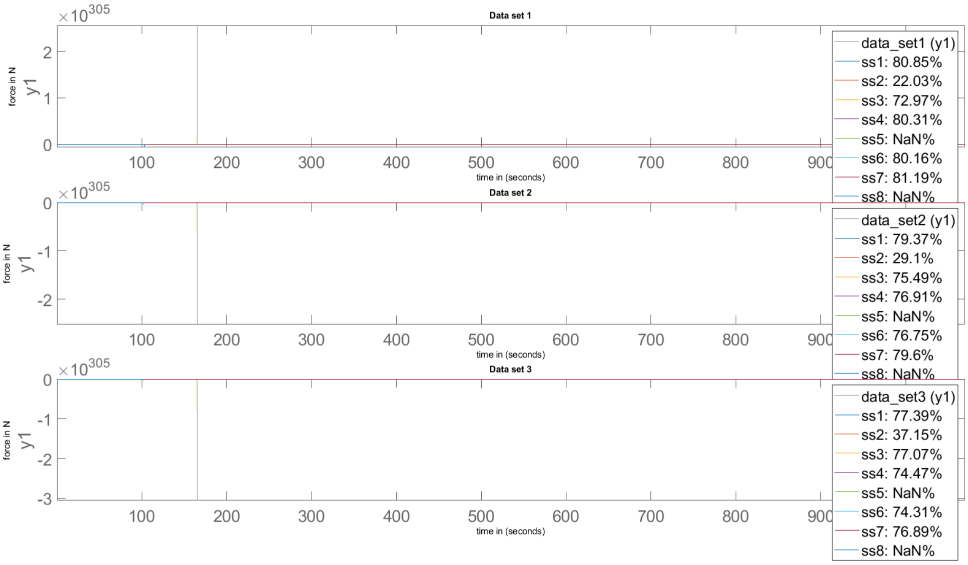


Figure A-12: Performance of state space models with different orders.

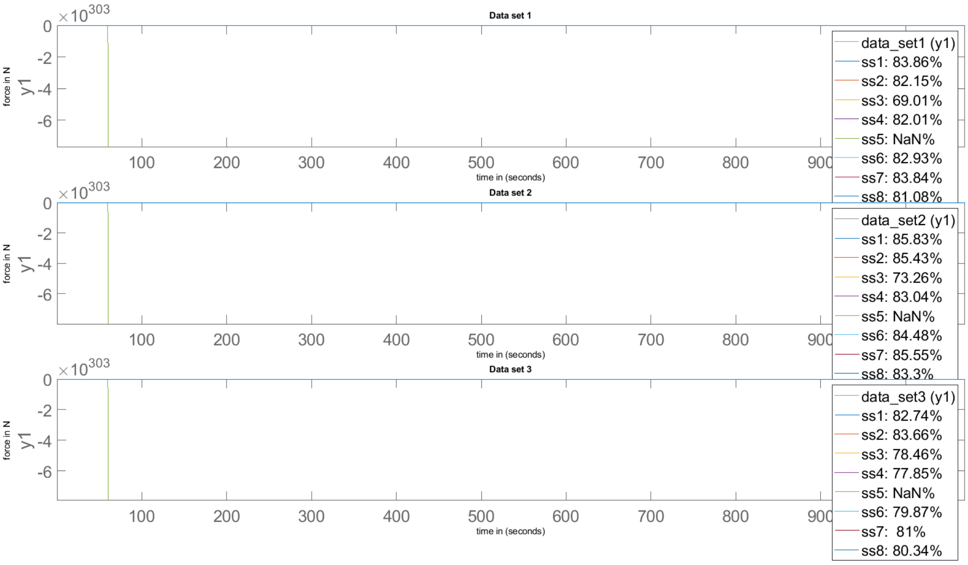


Figure A-13: Performance of state space models with different orders.

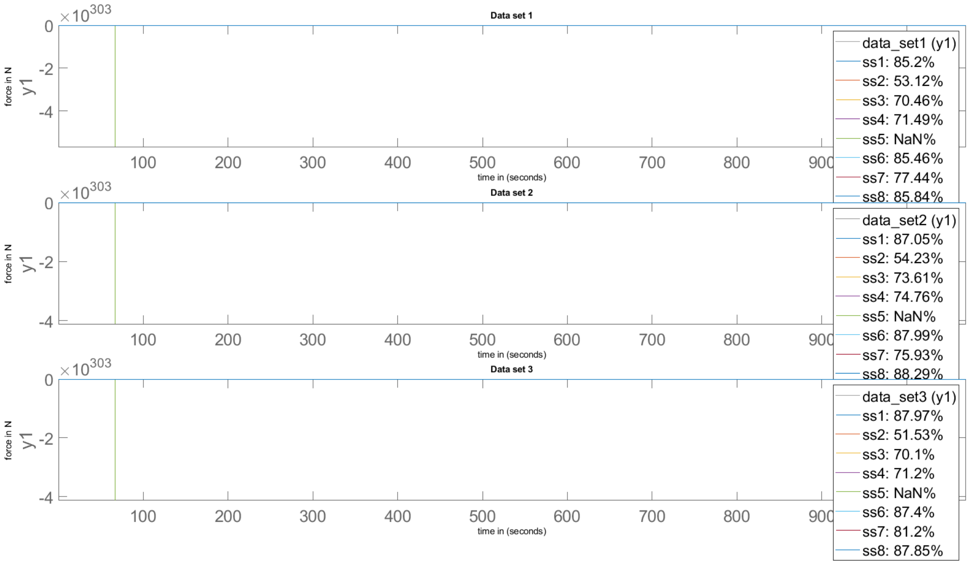


Figure A-14: Performance of state space models with different orders.

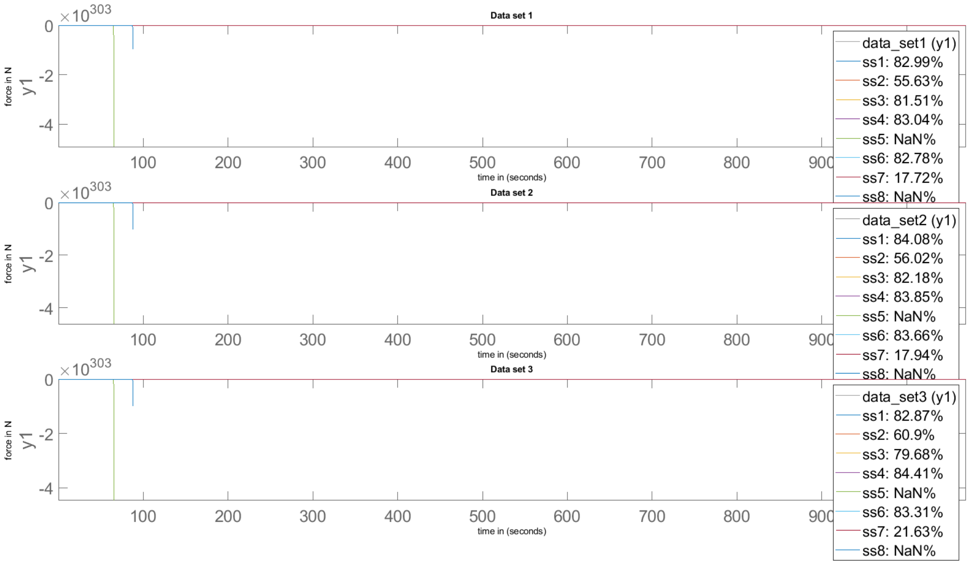


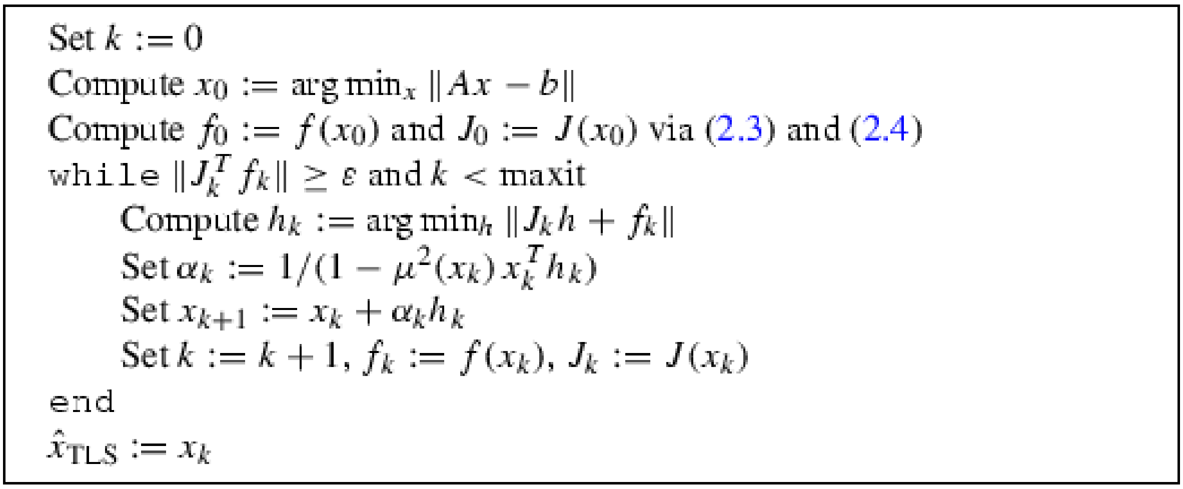
Figure A-15: Performance of state space models with different orders.

Appendix B

Least squares algorithms

B-1 The subspace Gauss-Newton

The subspace Gauss-Newton algorithm is given in Figure B-1.



```
Set $k := 0$
Compute $x_0 := \arg\min_x \|Ax - b\|$
Compute $f_0 := f(x_0)$ and $J_0 := J(x_0)$ via (2.3) and (2.4)
while $\|J_k^T f_k\| \geq \varepsilon$ and $k <$ maxit
    Compute $h_k := \arg\min_h \|J_k h + f_k\|$
    Set $\alpha_k := 1/(1 - \mu^2(x_k) x_k^T h_k)$
    Set $x_{k+1} := x_k + \alpha_k h_k$
    Set $k := k + 1$, $f_k := f(x_k)$, $J_k := J(x_k)$
end
$\hat{x}_{\mathrm{TLS}} := x_k$
```

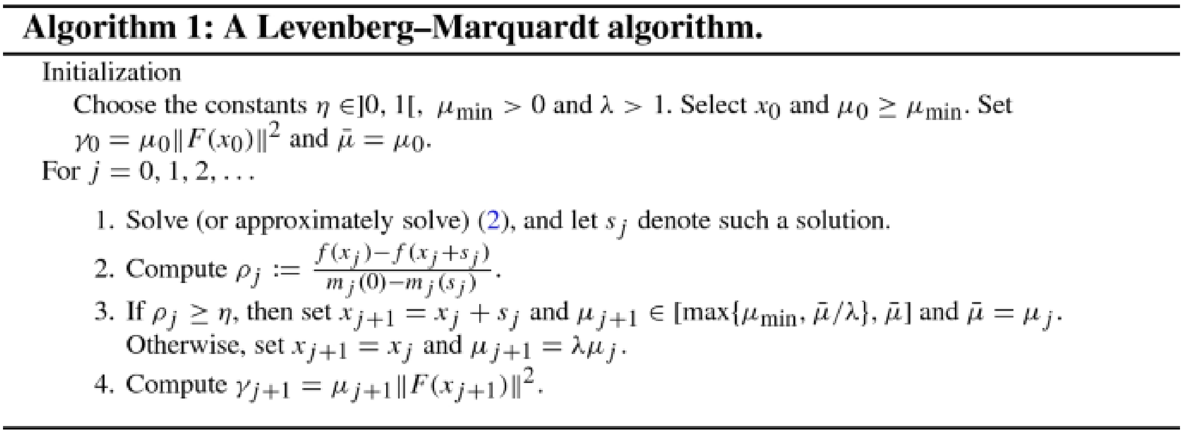
Figure B-1: An iteration of the Gauss-Newton algorithm (Fasino and Fazzi [2018])

B-2 The Levenberg-Marquardt

The Levenberg-Marquardt algorithm is given in Figure B-2.

B-3 The adaptive subspace Gauss-Newton search

Similar to Appendix B-1, but adaptive.

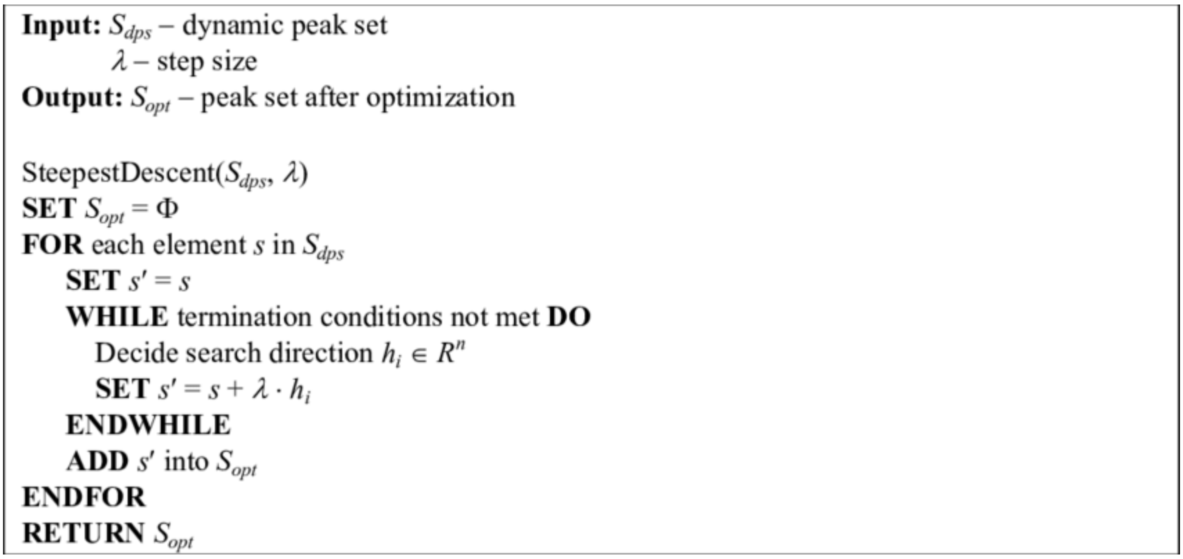


```
Algorithm 1: A Levenberg–Marquardt algorithm.
Initialization
    Choose the constants $\eta \in ]0, 1[$, $\mu_{\min} > 0$ and $\lambda > 1$. Select $x_0$ and $\mu_0 \geq \mu_{\min}$. Set
    $\gamma_0 = \mu_0 \|F(x_0)\|^2$ and $\bar{\mu} = \mu_0$.
For $j = 0, 1, 2, \ldots$
    1. Solve (or approximately solve) (2), and let $s_j$ denote such a solution.
    2. Compute $\rho_j := \frac{f(x_j)-f(x_j+s_j)}{m_j(0)-m_j(s_j)}$.
    3. If $\rho_j \geq \eta$, then set $x_{j+1} = x_j + s_j$ and $\mu_{j+1} \in [\max\{\mu_{\min}, \bar{\mu}/\lambda\}, \bar{\mu}]$ and $\bar{\mu} = \mu_j$.
       Otherwise, set $x_{j+1} = x_j$ and $\mu_{j+1} = \lambda\mu_j$.
    4. Compute $\gamma_{j+1} = \mu_{j+1}\|F(x_{j+1})\|^2$.
```

Figure B-2: The Levenberg-Marquardt algorithm (Bergou et al. [2020])

B-4 The steepest descent

The steepest descent algorithm is given in Figure B-3.



```
Input: $S_{dps}$ – dynamic peak set
       $\lambda$ – step size
Output: $S_{opt}$ – peak set after optimization

SteepestDescent($S_{dps}$, $\lambda$)
SET $S_{opt} = \Phi$
FOR each element $s$ in $S_{dps}$
    SET $s' = s$
    WHILE termination conditions not met DO
        Decide search direction $h_i \in R^n$
        SET $s' = s + \lambda \cdot h_i$
    ENDWHILE
    ADD $s'$ into $S_{opt}$
ENDFOR
RETURN $S_{opt}$
```

Figure B-3: The steepest descent algorithm (Wang [2007])

Bibliography

A Abbas and J Zhao. A physics based model for twisted and coiled actuator. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

T Arakawa, K Takagi, K Tahara, and K Asaka. Position control of fishing line artificial muscles (coiled polymer actuators) from nylon thread. In *SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring*, 2016.

E H Bergou, Y Diouane, and V Kungurtsev. Convergence anc complexity analyis of a levenberg-marquadt algorithm for inverse problems. *Journal of Optimization Theory and Applications*, 185:927–944, 2020.

A Cherubini, G Moretti, R Vertechy, and M Fontana. Experimental characterization of thermally-activated artificial muscles based on coiled nylon fishing line. *AIP Advances*, 5, 2015.

D Fasino and A Fazzi. A gauss-newton iteration for total least squares problems. *BIT Numerical Mathematics*, 58:281–299, 2018.

C Haines, M Lima, N Li, G Spinks, J Foroughi, J Madden, S Kim, S Fang, M De Andrade, F Göktepe, O Göktepe, S Mirvakili, S Naficy, X Lepró, J Oh, M Kozlov, S Kim, X Xu, B Swedlove, G Wallace, and R Baughman. Artificial muscles from fishing line and sewing thread. *Science*, 343:868–872, February 2014.

M Jafarzadeh, N Gans, and Tadesse Y. Control of tcp muscles using takagi-sugeno-kang fuzzy inference system. *Mechatronics*, 53:124–139, 2018.

K Masuya, S Ono, K Takagi, and K Tahara. Feedforward control of twisted and coiled polymer actuator based on a macroscopic nonlinear model focusing on energy. *IEEE Robotics and Automation Letters*, 3(3):1824–1831, July 2018.

D Roylance. Engineering viscoelasticity. 2001.

R.A. Serway and J.W. Jewett. *Physics for Scientists and Engineers*. Cengage Learning, 9 edition, 2013.

S Shafari and G Li. A multiscale approach for modeling actuation response of polymeric artificial muscles. *Soft Matter*, 11(19):3833–3843, March 2015.

A Simeonov, T Henderson, Z Lan, G Sundar, A Factor, J Zhang, and M Yip. Bundled supercoiled polymer artificial muscles: design, characterization, and modeling. *IEEE Robotics and Automation Letters*, 3(3):1671–1678, July 2018.

L Sutton, H Moein, A Rafiee, J Madden, and C Menon. Design of an assistive wrist orthosis using conductive nylon actuators. In *6th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2016.

M Suzuki and N Kamamichi. Simple controller design based on internal model control for twisted and coiled polymer actuator. *Actuators*, 7(33), June 2018.

H J Tulleken. Generalized binary noise test-signal concept for improved identification-experiment design. *Automatica*, 26(1):37–49, 1990.

J Van Der Weijde, E Vlasblom, P Dobbe, H Vallery, and M Fritschi. Force sensing for compliant actuators using coil spring inductance. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

J Van Der Weijde, B Smit, M Fritschi, C Van De Kamp, and H Vallery. Self-sensing of deflection, force and temperature for joule-heated twisted and coiled polymer muscles via electrical impedance. *IEEE/ASME Transactions on Mechatronics*, December 2016.

J Van der Weijde, H Vallery, and R Babuska. Closed-loop control through self-sensing of a joule-heated twisted and coiled polymer muscle. *Soft Robotics*, 2019.

L Wang. A comparison of three fitness prediction strategies for interactive genetic algorithms. *Journal of Information Science and Engineering*, 23(2):605–616, 2007.

L Wu and Y Tadesse. Modeling of the electrical resistance of tcp muscle. In *Proceedings of the ASME 2017 International Mechanical Engineering Congress and Exposition*, 2017.

L Wu, M de Andrade, R Rome, C Haines, M Lima, R Baughman, and Y Tadesse. Nylon-muscle-actuated robotic finger. In *Proceedings of SPIE: Active and Passive Smart Structures and Integrated Systems 9431*, 2015.

M Yip and G Niemeyer. High-performance robotic muscles from conductive nylon sewing thread. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

M Yip and G Niemeyer. On the control and properties of supercoiled polymer artificial muscles. *IEEE Transactions on Robotics*, 33(3):689–699, June 2017.

P. Young and A. Jakeman. Refined instrumental variable methods of recursive time-series analysis part iii. extensions. *International Journal of Control*, (4):741–64, 1980.

J Zhang, A Iyer, and M Yip. Modeling and inverse compensation of hysteresis in supercoiled polymer artificial muscles. *IEEE Robotics and Automation Letters*, 2(2):773–780, April 2017.

Glossary

List of Acronyms

TCPM	twisted coiled polymer muscle
MD	muscle drive
UTM	Universal Testing Machine
RMSE	Root Mean Square Error
NRMSE	Normalized Root Mean Square Error

List of Symbols

x	deflection
L	decay time, a measure of inductance
T	(contribution of) temperature
F	force
μ_0	magnetic permeability
l	length
r	radius
κ_P	coefficient of conductive heating
κ_c	coefficient of convective cooling
P	power
D	duty cycle
R_m	resistance measured at ends of TCPM
R_b	resistance measured connectors to power supply
U_b	voltage applied at connectors
k	stiffness
c	damping coefficient
f	frequency
ϕ	phase
τ	step time

h	Heaviside step function
b	thermal constant
C_{th}	Thermal mass
λ	thermal conductivity
e	error
K	controller gain