MASTER

Control of a Hysteretic Walking Piezo Actuator

Elmensdorp, Yves J.M.

Award date:
2021

EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Control Of A Hysteretic Walking Piezo Actuator

Master's thesis

Student:	Y.J.M. Elmensdorp
IDNR:	1393944
Master:	Systems and Control
Department:	Mechanical Engineering
Research group:	Control Systems Technology
Report ID:	CST2021.060
Supervisors:	ir. N. Strijbosch
	dr. ir. K. Tiels
	prof. dr. ir. T Oomen

Eindhoven, September 17 2021

[This page is intentionally left blank]

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date 5/9/21

Name Y.J.M. Elmensdorp

ID-number 1393944

Signature

Insert this document in your Master Thesis report (2nd page) and submit it on Sharepoint

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

[This page is intentionally left blank]

Preface

This report is the outcome of a graduation project for the Master's degree in System & Control at Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands. This project is carried out within the Control System Technology (CST) group at the Mechanical Engineering department. Supervision during this thesis is performed by Nard Strijbosch, a Ph.D. candidate at the TU/e.

This project is focusing on the modeling and control of the hysteresis phenomena that occur in the actuation of a piezoelectric based actuator. Furthermore, the hysteresis modeling and its extensions/variations are based upon the Prandtl-Ishlinski model. During this project an experimental setup consisting out of an actuator, sensors, and processing units where available.

With the completion of this research, a chapter in my life is concluded, both literally and figuratively. Looking back at my research, I can say that I can be proud of the result that has been achieved. During this period of more than half a year, I was able to broaden my knowledge both theoretically and practically. Closing this chapter would not have been possible without the help of several people.

First, I would like to thank Nard Strijbosch for the guidance during my research, Tom Oomen for being my mentor and the accompanying advice, Koen Tiels for his great knowledge about Neural Networks and everyone who has been involved with interest, patience, and ideas over the past half a year.

Finally, I would like to thank my girlfriend, father, and friends for the considerable distraction and relaxation when this was needed. I would also like to thank my grandfather, who unfortunately is not able to see my graduation anymore, for his support and interest during my study. After having completed almost the entire Dutch education system, read VMBO-Kader → MBO-4 → HBO bachelor → WO master, it is eventually really time to grow up and start looking for a job.

I hope you enjoy your reading!

Yves Elmensdorp
Eindhoven, 17 September 2021

[This page is intentionally left blank]

Abstract

Actuators provide high precision control, rapid response, and low energy consumption in the fields of semiconductor manufacturing, and electron microscopy. Common actuator types make use of guides or pistons to transfer energy. Therefore, the use of these kinds of actuators is not possible in applications with for example a cleanroom environment. In these cases, piezo actuators are used to comply with the specifications of the precision system. Piezo actuators make use of piezo elements that are actuated with a waveform to generate a static walking movement. This walking movement is translated into small steps (10^{-6} [m]) and therefore are ideally for positioning objects with high precision. Shear and clamp elements are actuated with a waveform that are constructed such that the timing of the elements results in a continuous walking movement mechanical piece. Moreover, piezo actuators convert electrical energy to mechanical energy through the piezoelectric effect. Therefore, piezo elements are classified as ferroelectric material and therefore hold performance decreasing phenomena such as creep and hysteresis. In addition, open-loop experiments showed that performance is also affected by the type of input waveform and the direction of actuation.

This research is focused on the modeling and compensation of the hysteresis phenomena based on the Prandtl-Ishlinskii hysteresis model. Compensation is achieved by the use of a feed-forward compensator based on the inverse Prandtl-Ishlinskii model. As an extension, variations on the classical Prandtl-Ishlinskii model are proposed. This included the introduction of a generalized envelope function for the play operator, a memory element adaption, and a neural network adaption of the PI model. The compensators are implemented and tested on the shear elements for performance analysis. Experimental results showed that the proposed compensators had similar performance. However, the generalized Prandtl-Ishlinskii model performed best out of the constructed models. On the contrary, the memory element model consists out of half the number of elements and would therefore outperform the other models when equalizing the number of elements.

The generalized Prandtl-Ishlinskii compensator is implemented in combination with an iterative learning controller with basis functions. There are two sets of basis functions that are chosen based on prior knowledge of the systems response. The first set is used to scale the shear waveforms. The second set is based on the Fourier series of the measured disturbances. Experimental results of the system in walking behavior showed a significant increase in performance due to the learning controller and the hysteresis compensator compared to the uncontrolled and uncompensated situation. However, RMS error fluctuation occurs when outside the 1 [Hz] hysteresis compensator region which is an effect of the compensator not being able to handle rate-dependent hysteresis behavior. Therefore, the analysis of the system with hysteresis compensation, learning controller, and varying reference input could not be executed. In the end, the system response with a 1 [Hz] drive frequency for the situations with ILC and hysteresis compensation for both mover directions result in a significant error decrease for the positive and negative walking directions compared to the uncontrolled and uncompensated situation.

[This page is intentionally left blank]

Contents

List of Figures

List of Tables

Acronyms

ANN	Artificial Neural Network.
CNN	Convolution Neural Network.
DC	Direct Current.
GPI	Generalized Prandtl-Ishlinskii.
GRU	Gated Recurrent Unit.
ILC	Iterative Learning Control.
LBFGS	Broyden–Fletcher–Goldfarb–Shanno.
LSTM	Long Short-Term Memory.
MSE	Mean Square Error.
NN	Neural Network.
PI	Prandtl-Ishlinskii.
PI-NN	Prandtl-Ishlinskii Neural Network.
ReLU	Rectified Linear Unit.
RMS	Root Mean Square.
RNN	Recurrent Neural Network.
RT	Real-Time.

[This page is intentionally left blank]

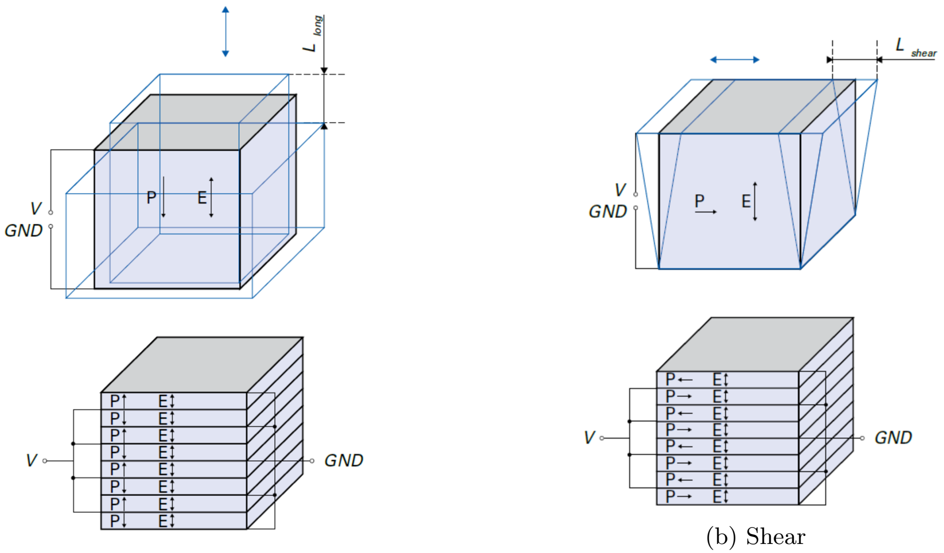
Chapter 1

A Brief Introduction on Walking Piezo Actuators

Actuators must provide high precision control, rapid response, and low energy consumption in the fields of industrial machinery, semiconductor manufacturing, and electron microscopic. A common method in precision motion control is to make use of servo or linear motors in combination with an advanced controller. Most of the common actuator types make use of guides or pistons to transfer energy. Therefore, the use of these kinds of actuators is not possible in applications where, for example, there is a vacuum, cleanroom environment, or nanometer precision. In these cases, piezo actuators are used to comply with the specifications of the precision system. In this chapter, an introduction on the piezo actuator, its internal and external elements, and the method of actuation are discussed.

1.1 Piezoelectric Elements

piezo actuators make use of piezo elements that are actuated in a particular order to generate a movement. Furthermore, piezo actuators convert electrical energy to mechanical energy through the piezoelectric effect and therefore have no moving parts that generate friction or wear. Common applications for piezo actuators are in XY-stages [1], vibrating feeders [2], electron microscopes [3], and surgery robots [4]. A piezo actuator consists out of multiple stacked ceramic piezo elements that expand or contract depending on the electrical charge that is applied. Furthermore, the deformation and its direction are linearly related to the electrical charge and the polarization of the material. Therefore, piezo elements are classified as ferroelectric material and hence are affected by behavioral interference such as creep and hysteresis. Depending on the polarization there are two types of directions: longitude or shear. Figure 1.1a shows the working principle of the longitude element. Here, the electric field in the ceramic layer is applied parallel to the direction of polarization. Figure 1.1b shows the working principle of the shear element. Here, the electric field in the ceramic layer is applied orthogonal to the direction of polarization. Both polarizations result in a displacement in the direction of the polarization.



(a) Longitude

(b) Shear

Figure 1.1: Piezo element polarization directions. Top indicate the deformation, bottom indicate the stacked piezo element layers [5].

1.2 Walking Piezo Actuator

Walking piezo actuators are actuators that make use of piezo elements to produce a linear motion. Walking piezo actuators (or piezo stepper) make use of the inverse piezoelectric effect which results in a contraction or extension of the material. The piezo actuator used in this research consists of two types of piezo elements: shear and clamp (longitude). The entire actuator consists of a protecting housing, two sets of clamp and shear elements, and a mover bar. Figure 1.2 shows an overview of the used piezo stepper with its external components. Figure 1.3a and 1.3b show a schematic overview of the internal principle in front and side view. The shear and clamp elements are depicted respectively as S and C and the actuation sets are depicted as 1 (🟦) and 2 (🟥). The elements are actuated by a high voltage Direct Current (DC) input waveform that is generated by an analog controlled voltage amplifier. By constructing corresponding clamp and shear waveforms, a continuous stroke, and therefore a walking behavior can be obtained while maintaining high accuracy.

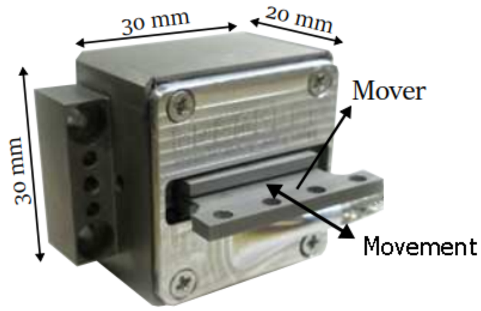
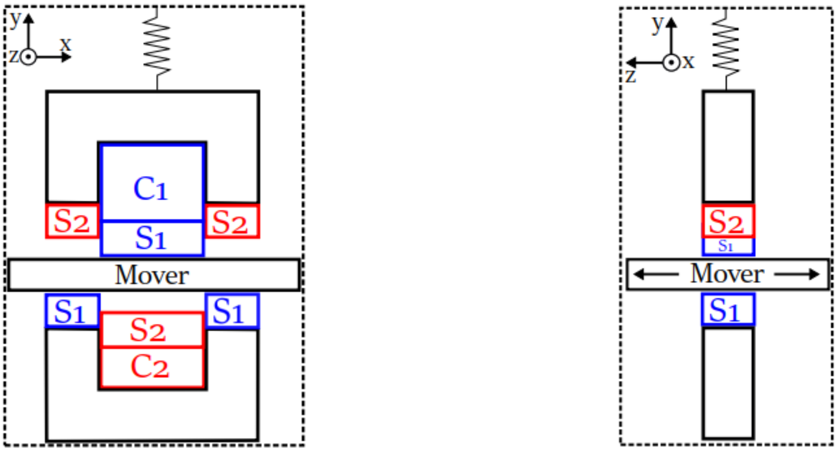


Figure 1.2: Overview of the piezo actuator with its dimensions and external components [6].



(a) Front view of actuator

(b) Side view of actuator

Figure 1.3: Schematic working principle of the piezo stepper with (a) front view of internal principle and (b) side view of internal principle where S indicates a shear element, C a Clamp element and the number 1 (▬) or 2 (▬) the element set [6].

1.3 Input Waveform

The clamp and shear elements are actuated with a generated waveform such that the shear elements translate a ”walking” behavior on the surface of the mover and therefore translate the mover in a fixed axis direction. Furthermore, the waveforms are constructed in the commutation angle domain α [rad] [7] and defined periodic with a period from 0 to 2π. The first derivative with respect to time of the commutation angle (i.e. angular velocity) determines the number of waveforms per second. Therefore, the commutation angle is derived from the drive frequency f_α [Hz] and defined by (1.1).

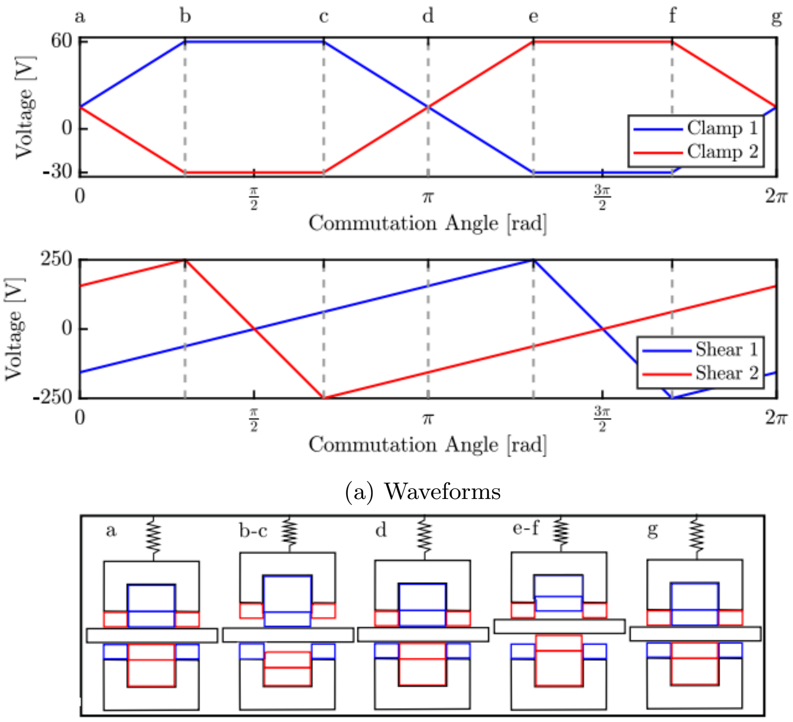
$$\alpha(t) = 2\pi \int_0^t f_\alpha(\tau) d\tau \tag{1.1}$$

The shear waveforms are constructed in such a way that the derivatives are equal for all α where the corresponding clamp elements are in contact with the mover. By doing so, a linear relationship is obtained between the commutation angle and the mover displacement. During walking operation, the shear elements are ”reset” to their minimum position (depending on the direction of walking) to achieve a continuous motion of the mover since the movement of a shear element is limited. Since the shear elements move with the same velocity, the input derivative of the shear sets can be defined by (1.2) where the bounding conditions are the regions where the corresponding clamps C1 and C2 do not make contact with the mover.

$$\frac{\delta u_j(\alpha)}{\delta \alpha} = \begin{cases} \frac{\delta u_{S1}(\alpha)}{\delta \alpha} & \text{if } \alpha \in \left[\frac{9\pi}{20}, \frac{11\pi}{20}\right] \\ \frac{\delta u_{S2}(\alpha)}{\delta \alpha} & \text{if } \alpha \in \left[\frac{29\pi}{20}, \frac{31\pi}{20}\right] \\ \frac{\delta u_{S1}(\alpha)}{\delta \alpha} = \frac{\delta u_{S2}(\alpha)}{\delta \alpha} & \text{otherwise} \end{cases} \tag{1.2}$$

Figure 1.4 illustrates the stepping procedure for a single step. Here, 1.4a illustrates the generated waveform of a single step in the commutation angle domain and 1.4b the piezo element behavior per stage for each actuation set. Furthermore, a positive and negative voltage refers respectively to an extended and retracted piezo element. The regions between the letters **a-g** indicate the different actuation stages for a single step:

- **a**: Start of a drive cycle i.e. commutation angle equals zero. All shear sets make contact with the mover which results into a takeover moment between the shear sets. The derivative with respect to the commutation angle of the shear voltages, $\frac{dV}{d\alpha}$ [V/rad], of both shear groups is equal and therefore the movement velocity.
- **a-b**: Clamp 1 expands to its maximum length and clamp 2 retracts to its minimal length.
- **b-c**: Clamp 1 is fully extended and makes contact with the mover. Clamp 2 is fully retracted and shear set 2 is reset to the initial position i.e. maximum stretched position in opposite moving direction.
- **c-e**: Clamp 1 is retracting and clamp 2 is extending.
- **d**: As with stage (a), there is takeover moment and the velocity of the shear groups are equal.
- **e-f**: Clamp 2 is fully extended and makes contact with the mover. Clamp 1 is fully retracted and shear set 1 is reset to the initial position.
- **f-g**: Clamp 1 expands to its maximum length and clamp 2 retracts to its minimal length.
- **g**: End of a drive cycle i.e. commutation angle equals 2π and resets the cycle.



(a) Waveforms

(b) Drive Cycle

Figure 1.4: Drive cycle per different stage of a single step of movement of the piezo stepper. (a) Clamping and shear waveforms as function of commutation angle. (b) Schematic front view of the piezo stepper indicating the different stages in the drive cycle. [6].

1.4 Problem Definition

A disadvantage of using piezo-based actuators is that piezo material has the property of introducing hysteresis. Hysteresis is a non-linear phenomenon that occurs in ferroelectric and ferromagnetic materials such as piezo material. As a result, forward and backward displacement do not coincide and therefore limit the performance of the actuator. Because of this, it is desirable to compensate for this hysteresis behavior. Furthermore, previous research on the experimental setup showed that the system is subjected to periodic disturbances. Since this limits the performance of the actuator, an extra controller should be implemented that is capable of reducing periodic disturbances. Therefore, this research will focus on the modeling and compensation of the hysteresis phenomena, and in particular with the Prandtl-Ishlinskii hysteresis model. In addition, a design for a controller is proposed to reduce periodic disturbances acting on the system. This research can be split into the following parts:

1. Investigation of the open-loop system behavior and occurring phenomena.
2. Hysteresis modeling and compensation for a piezo element.
3. Development of a learning controller framework that is able to reduce periodic disturbances.
4. Integration of the developed controller with and without hysteresis compensation on the experimental setup.

1.5 Research Outline

The research goal defined by the previous section is executed in order with particular steps and discussed in their respective chapter. The results of this research are represented in the following structure:

- Chapter 2 presents a behavioral background of the experimental setup. Here, system behavior and properties such as open-loop responses, periodic disturbances, and hysteresis are discussed. Furthermore, the control structure and setup construction are discussed.
- Chapter 3 presents the modeling and compensation of the hysteresis phenomena in a piezo element with the Prandtl-Ishlinskii model. In addition, multiple extensions on the models are discussed and tested on the experimental setup.
- Chapter 4 presents the design and implementation of a iterative learning controller with basis functions. Furthermore, the controller is tested on the experimental setup in walking behavior with and without hysteresis compensation.
- Chapter 5 concludes the findings within the research. Furthermore, a recommendation is discussed for future research on the research problem.

Chapter 2

Experimental Setup and System Behavior

The aim of this chapter is to obtain a better understanding of the experimental setup that is used within this research and its behavior when actuated. In Section 2.1 the components of the experimental setup such as the peripherals, software, and actuator are discussed. In Section 2.2 the actuation behavior in open-loop is discussed which gives a better understanding of performance limiting behavior within the system.

2.1 Experimental Setup and its Components

An experimental setup was available during this project in order to obtain data in a real-life environment. Figure 2.1 shows the piezo stepper of the experimental setup with its components. The experimental setup consists out of a piezo stepper (actuator), a linear encoder (readhead) for mover position measurements, a voltage amplifier, and a Speedgoat Real-Time (RT) controller with analog IO's to control the setup. The Speedgoat controller is connected to a desktop PC that hosts the simulation program. The software environment used consists out of simulation software MATLAB and Simulink. Figure 2.2 shows the high-level overview of the experimental setup. Moreover, the encoder signals have an additional error due to misalignment and mechanical tolerances of the construction. Therefore, the encoder is calibrated by a so called Lissajous calibration correction. The full calibration procedure is explained in Appendix A.

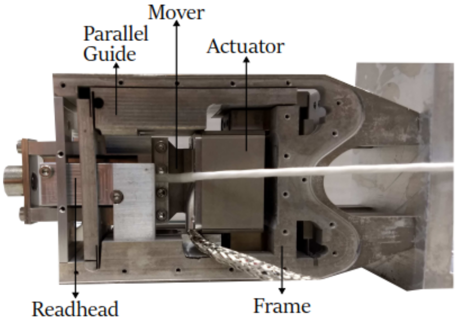


Figure 2.1: Piezo stepper of experimental setup with its components [6].

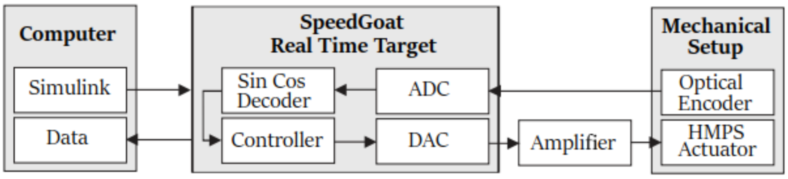


Figure 2.2: High level overview of the experimental setup [6].

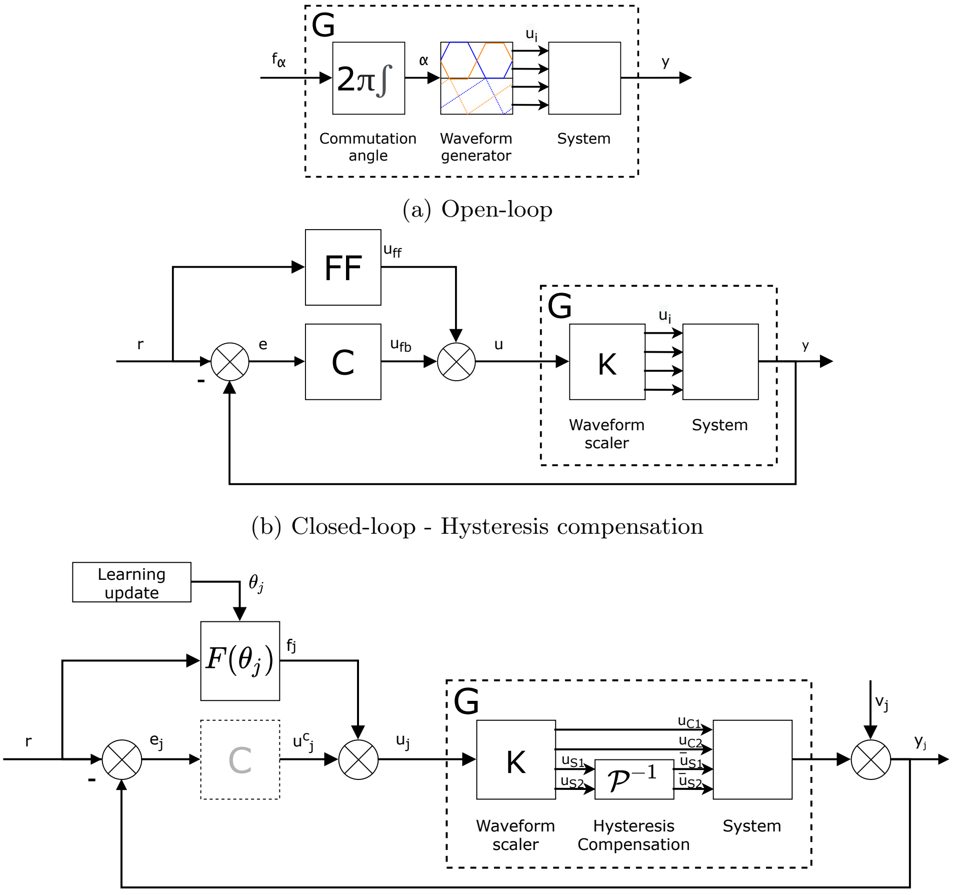
2.1.1 Control Architecture

The control architecture is the internal structure used for controlling and experimenting with the system. Within this project, there are two types of control structures used:

1. **Open-loop structure**
   The open-loop structure as illustrated in Figure 2.3a is used for all initial non-controlled experiments. These experiments are walking experiments, hysteresis experiments, and creep experiments. The open-loop structure has drive frequency f_α [Hz] as the input and mover position y [m] as its output. Furthermore, the plant is represented by a series of actions: drive frequency to commutation angle α [rad] conversion (1.1), waveform generating, and the actuation behavior. The waveform generator is defined by a trapezoidal and a sawtooth waveform for respectively the clamps and the shears.

2. **closed-loop structure - Hysteresis compensation**
   The closed-loop structure as illustrated in Figure 2.3b is used for hysteresis compensation on the shear elements. The structure consists out of a stabilizing feedback controller, feedforward controller, and a plant. The feedback controller consists out of a low-pass filter and a gain. The plant used for the closed-loop structure does not make use of the commutation angle domain since the input waveform is determined by the controllers. Therefore, only a waveform scaler is used. The closed-loop structure has a reference position r [m] as input and mover position y [m] as its output. Furthermore, the tracking error is depicted as e, the feedback controller output as u_{fb}, the feedforward controller output as u_{ff}, and the plant input as u. The feedforward controller is used for hysteresis compensation which will be discussed in chapter 3

3. **Closed-loop structure - Iterative learning control**
   The closed-loop structure as illustrated in Figure 2.3c is used for iterative learning control with and without hysteresis compensation. The structure consists out of a iterative learning controller with basis functions $F(\theta_j)$ and a plant G with incorporated hysteresis compensation $\mathcal{P}^{-1}$. The closed-loop structure has a reference position r [m] as input and mover position y [m] as its output. Furthermore, the tracking error is depicted as e_j, the basis function output as f_j, and the disturbances as v_j. The trial index is denoted as j. The plant used for the closed-loop structure does not make use of the commutation angle domain since the input waveform is determined by the controller. Therefore, only a waveform scaler is used. The construction of the iterative learning controller and its basis functions will be discussed in Chapter 4.



(a) Open-loop

(b) Closed-loop - Hysteresis compensation

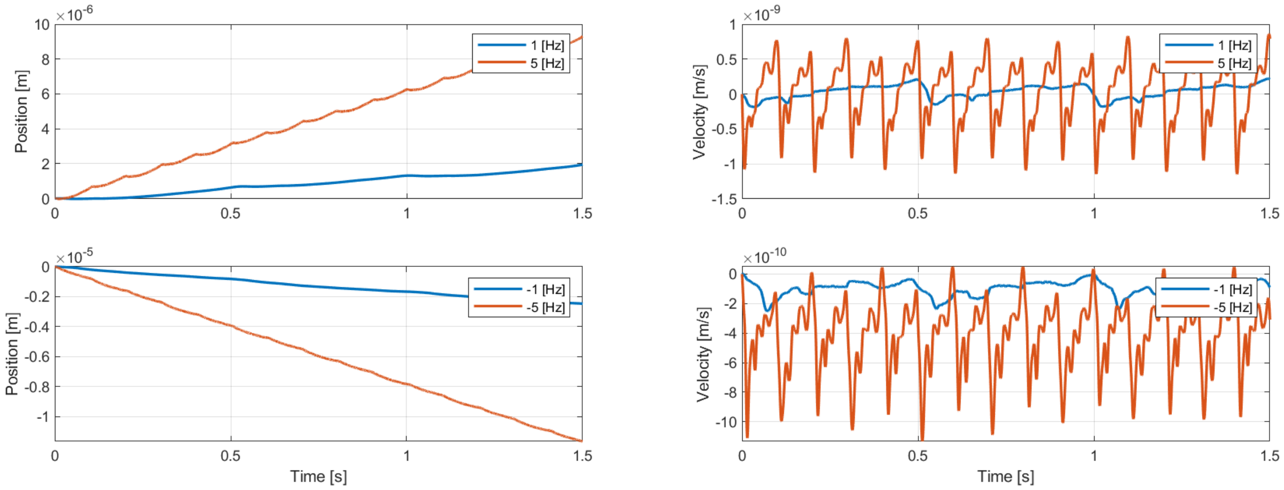
(c) Closed-loop - Iterative learning control

Figure 2.3: Block scheme of control architecture in (a) open-loop experiments, (b) closed-loop experiments with feedforward hysteresis compensation, and (c) closed-loop experiments with iterative learning control and hysteresis compensation.

2.2 Open-Loop Experiments

A walking experiment is carried out to investigate the open-loop ”walking” behavior of the piezo stepper. In this experiment, the actuator is driven with a drive frequency of $f_\alpha \in \{1, 5, 10, 20\}$ [Hz] and there negative counterparts. The step size of the actuator is approximately 3×10^{-6} [m].

Figure 2.4a shows the filtered output position of the mover for $f_\alpha \in \{1, 5, -1, -5\}$ [Hz]. A position output in the form of a ramp is expected since there is a linear relationship between the input voltage on the piezo element and the output displacement. However, as can be seen in the figure, disturbances are observed twice every period. This is a result due to the hysteresis phenomena that occurs within the piezoelectric material. This behavior can also be seen in Figure 2.4b at the minima and maxima points on the filtered velocity curves. Moreover, it can be observed from the negative frequencies, that the displacement behavior is different compared to the positive slope.



(a) Position (b) Velocity

Figure 2.4: Walking behavior for $f_\alpha \in \{1, 5, -1, -5\}$ [Hz] with (a) the position and (b) the velocity curve

Clamp liftoff & contact moments

Clamp liftoff and contact moments play an important role in the walking behavior of the piezo actuator. Liftoff is defined as the first moment in a drive cycle where the clamp stops having contact with the mover. Opposite to liftoff, contact is defined as the first moment in a drive cycle where the clamp comes in contact with the mover. A liftoff/contact experiment is executed in order to investigate the moments these actions occur where the exact moment is portrayed by the current clamp voltages. In this experiment, a clamp set is actuated with a low frequency sawtooth waveform while the corresponding shear set is actuated with a faster sinusoidal waveform. The other clamp and shear set are stationary while the experiment is executed. The experiment results can be read in Appendix B.1. The experiments showed that there is a difference between contact and liftoff voltages for different actuation frequencies. Furthermore, it is observed that there is a varying difference between the positive and negative frequency voltages. In most frequency cases the data shows that the contact and liftoff moments happen at a lower voltage. As a result that the clamp elements are longer in contact with the mover and therefore produce a longer stroke.

Creep

Creep is the phenomenon where the actuator drifts slowly after an increase of the input for a constant applied electric field [8]. Creep reacts after the dynamic response of the system and settles in a logarithmic shape over time. The amount of creep is determined by the size of the input difference on the piezo element. Experiments showed that the output position drifts in time for big input differences. Since the size of the input difference is small for the controlling applications, the effect of creep is negligible and therefore will not be modeled. The experiment results of the creep experiment can be read in Appendix B.2.

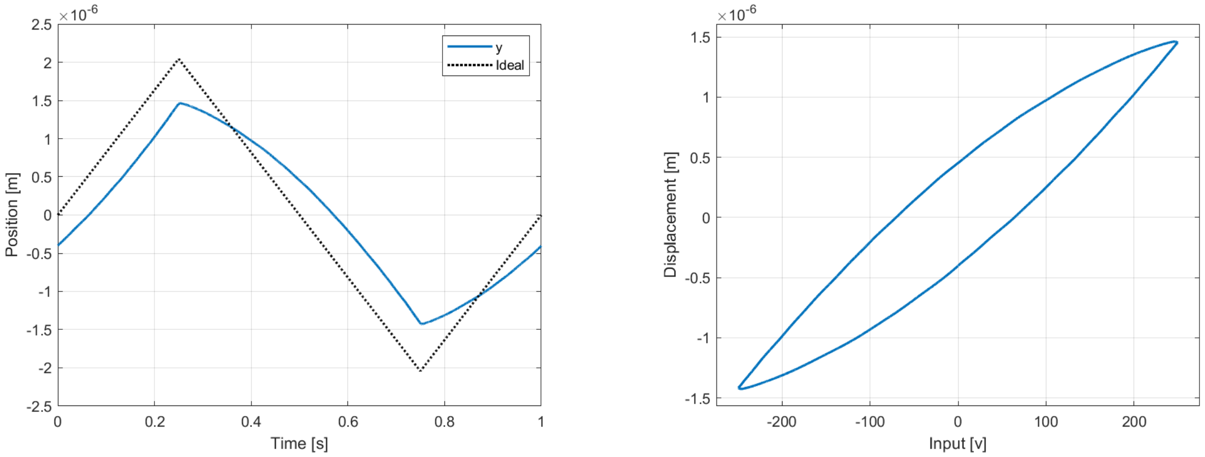
Input waveform type

The input waveform plays an important role in the motion performance of a piezo actuator. A piezo actuator can be driven with different waveforms based on the desired performance e.g. velocity, motion, and force. Three common basic waveform types [9] are: square wave, sawtooth, and sinusoidal. A sawtooth waveform, as shown in 1.4a for the shear elements, was used in previous research on the experimental setup [10][11]. Alternatively, waveforms can be utilized by use of model/data-based optimization [12], iterative learning-based [13], or force-based [14]. An experiment with a static sawtooth and sinusoidal shear waveform showed that the the position curve of the sinusoidal waveform results in a longer more bouncier stroke compared to the sawtooth waveform. This is an effect of the sinusoidal waveform slightly compensating for some hysteresis behavior that occurs in the piezo element. In addition, the liftoff and contact moments occur at a different shear voltage compared to the sawtooth waveform. This results into a longer contact time with the mover and therefore into a longer stroke. For the scope of this project only a sawtooth waveform is used. The experiment results of the waveform experiment can be read in Appendix B.3.

Chapter 3

Hysteresis Compensation for a Piezo Element

Hysteresis is a non-linear phenomenon that occurs in ferroelectric and ferromagnetic materials such as piezo material. Hysteresis occurs when a system state is dependent on its history. In the piezoelectric ceramic, this effect is due to the spontaneous polarization alignment of the dipoles [15][16]. When translating this to the experimental setup, the input voltage range will cause the output displacement of an element to be different for the same voltage value. Figure 3.1b shows the input vs. displacement curve for shear set 1 of the experimental setup. The figure shows that the forward displacement curve does not coincide with the backward displacement curve and therefore the hysteresis in the piezo stepper limits the performance of the actuator. Because of this, it is desirable to compensate for the hysteresis phenomena. There are multiple possible dependencies such as amplitude and rate dependencies which affect the behavior of the hysteresis phenomena. Since both dependencies are applicable to the system, both cases are investigated.



(a) Position

(b) Input vs. displacement

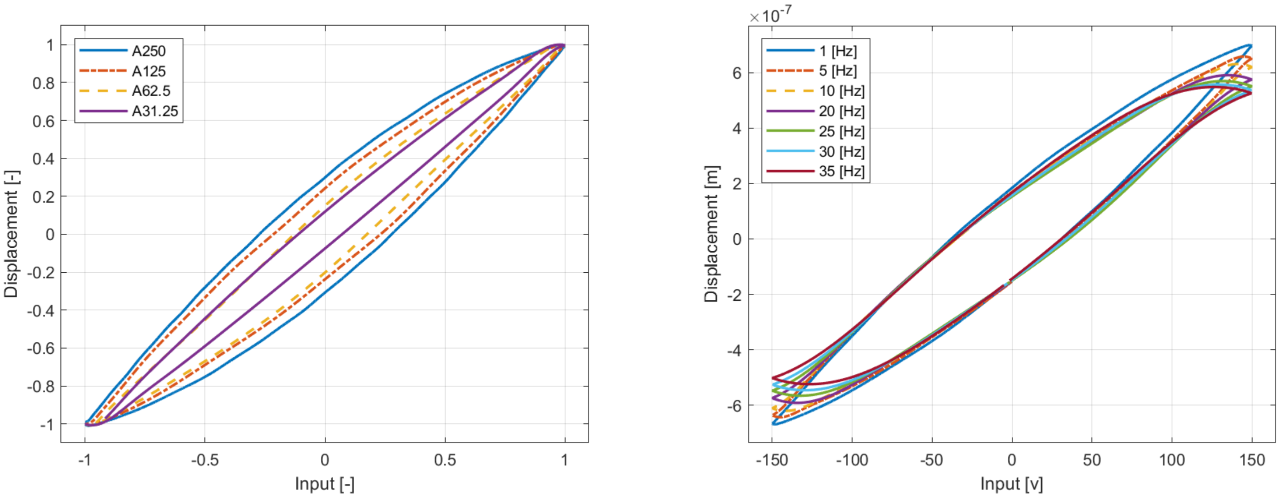
Figure 3.1: Visible hysteresis effect on experimental setup shear set 1 for (a) mover position where (⋯⋯) is the ideal trajectory and (b) input vs. Displacement.

Amplitude dependence

The amplitude dependency of the shear elements is investigated by actuating triangular waveforms with different amplitudes and a constant input rate. When a system is amplitude-dependent, it means that with varying input voltages and a constant rate, the amount of displacement in proportion to the input is the same. Figure 3.2a shows the resulting normalized hysteresis curve that occurs when exciting shear set 1 with amplitude values of $A \in \{31.25, 62.5, 125, 250\}$ [V]. It can be observed that for each waveform amplitude the curve has a different shape. Therefore, it can be concluded that the hysteresis behavior is amplitude-depended.

Rate dependence

The rate dependency of the shear elements is investigated by actuating triangular waveforms with a constant amplitude and varying step rates. Rate is defined as the time derivative of the input voltage applied to the shear element. When a system is rate-dependent, it means that the hysteresis behavior is dependent on the rate at which the minima and maxima are obtained. Figure 3.2b shows the resulting hysteresis curve that occurs when exciting shear set 1 with a constant amplitude of 150 [V] and changing step frequencies of $f_\alpha \in \{1, 5, 10, 20\}$ [Hz]. It can be observed that for increasing frequencies, the hysteresis curves at the minima and maxima bends in a clockwise direction. This results in a decreasing step size for increasing frequencies and thus concludes that the hysteresis behavior is rate-dependent.



(a) Normalized amplitude

(b) Rate

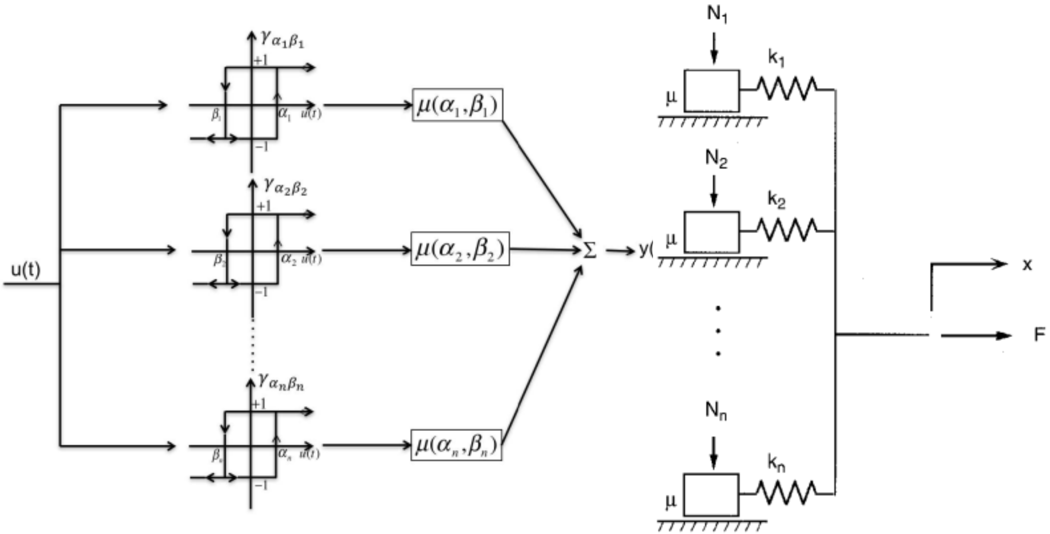
Figure 3.2: Hysteresis dependencies: (a) for signal with frequency $f_\alpha = 1$ [Hz] and amplitudes $A \in \{31.25, 62.5, 125, 250\}$ [V] and (b) for frequencies $f_\alpha \in \{1, 5, 10, 20\}$ [Hz] and $A = 150$ [V].

Because of this, it is desirable to compensate for the hysteresis so that the piezo stepper behaves in a linearly manner. The first step is to model the hysteresis behavior such that it matches the system response. The second step is to determine the inverse such that it can be used in feedforward control. Furthermore, the compensator should be able to deal with the shape-changing properties of these two dependencies. However, for simplicity and the choice of the modeling method, only the amplitude dependency is taken into account. Within the literature, much research is already performed on the modeling of hysteresis behavior. Hysteresis models can be divided into three main categories.

The first category, operator-based models, are based upon algebraic operators [17] such as play or stop operators that are combined to form a non-linear behavior. For example, a classical Preisach model is proposed in [18] and [19]. Figure 3.3a show the visual representation of the Preisach model. Here, operators are used for switching between +1 and -1 depending on their defined thresholds. The operators are multiplied by a weight and connected in parallel i.e. hysterons. The higher the amount of hysterons, the more accurate and computationally difficult the model gets. Furthermore, parameterization is obtained by determining the coefficients via data fitting based upon actuator data. A contribution to the Preisach model results in the Prandtl-Ishlinskii (PI) model, which is the main topic of this report and is discussed in Section 3.1.1. Both models are based upon play or stop operators. However, contrary to the Preisach model, the PI model does have an analytic inverse and is therefore direct compatible with feedforward compensation.

The second category, differential-based models, is based upon differential equations. For example, a Duhem model is proposed in [20]. The generalized form of the Duhem model consists of an ordinary differential equation that is related to a phenomenological approach. The model parameters are based upon experimental data and obtained by adjusting the parameters in such a way that the model output matches the experimental data. Differential-based models have a reduced amount of parameters compared to detailed operator-based models and use less computational power. However, this does not hold for operator-based models that have a small number of parameter arrays.

The third category, physics-based models, are based upon the physical parameters of materials. For example, a Generalized Maxwell Slip model of the hysteresis effect is proposed in [21] [22]. Figure 3.3b show the visual representation of the Maxwell slip model. Here, the hysteresis behavior is modeled by a parallel interconnection of ideal springs in combination with pure Coulomb friction i.e. an elasto-slide element. A second example is the use of a MEMS-element [23] to describe a hysteresis model. As with the Maxwell model, the MEMS-model is based upon the behavior of a mechanical spring element. In [11] the hysteresis effect is modeled as a MEMS-model based on the Ramberg-Osgood hysteresis model [24]. Physics-based models are harder to implement compared to differential and operator-based models due to their increased complexity and computational effort. However, parameters and model behavior of physics-based models are better related to the real-life environment.



(a) Operator-based

(b) Physics-based

Figure 3.3: Visual representation of hysteresis model types: (a) operator-based preisach model [25] and (b) physics-based Maxwell slip model [21].

3.1 Modeling of Hysteresis with the Prandtl-Ishlinskii Model and its Variations

The next sections discusses the theory of the PI model and its variations. Here, more insight is obtained for the classical PI model in Section 3.1.1, the introduction of generalized play operator which results in the Generalized Prandtl-Ishlinskii (GPI) model in Section 3.1.2, and the memory element extensions of a MEMS implementation of the PI model in Section 3.1.3. Furthermore, in Section 3.1.4 the PI model is implemented as more advanced structure in the form of a Neural Network (NN) adaptation.

3.1.1 Prandtl-Ishlinskii Model

The Prandtl-Ishlinskii model is an operator-based model which is a subset of the Preisach model. A PI model consists of a weighted superposition (density function) of play/stop operators which are characterized by their operator radius. In this thesis, only the play operator-based model is discussed. The mathematical definition of the PI model is described by:

$$\begin{aligned} y(t) &= \mathcal{P}[u](t) \\ &= p(r_0)u(t) + \int_0^R p(r)F_r[u](t)dr \end{aligned} \tag{3.1}$$

Here, r is the threshold (radius), p is the threshold weight, F_r is the play operator, and R is the number of thresholds. In literature, R is usually chosen as infinity since the weighting function p_r vanishes for large values of r. This results in a smooth hysteresis curve made out of infinite piecewise elements. Play operator F_r is defined as:

$$F_r\left[u\right](0) = f_r(u(0), 0) = y(0) \tag{3.2a}$$

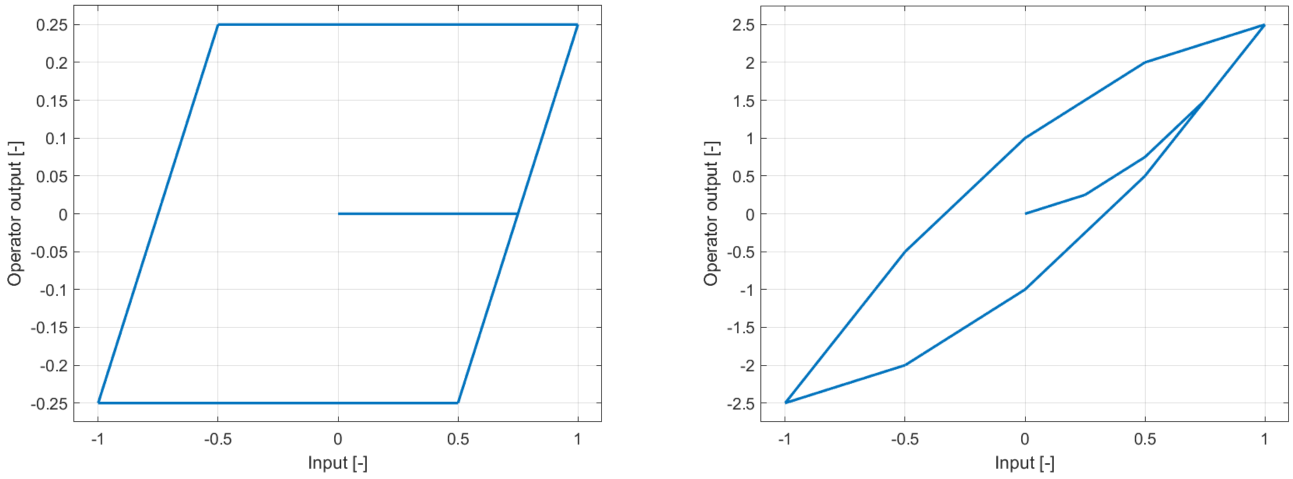
$$F_r\left[u\right](t) = f_r(u(t), F_r\left[u\right](t_i)) \tag{3.2b}$$

$$f_r(u, y) = \max(u - r, \min(u + r, y)) \tag{3.2c}$$

Where, u is a piecewise monotone input function and t_i is the previous time step for $t_i < t \leq t_{i+1}$ with $0 \leq i \leq N - 1$. The play operator acts as two mechanical elements with one dimension of freedom. Furthermore, the first element is bounded between the bounds of the second element. The bounds of the second element are defined as two times radius r. The position of the first element is controlled between the interior of the second element and is equal to the input value until a boundary is hit. Figure 3.4a illustrates a single play operator output with threshold 1 when a unit sawtooth input is applied. The integral in (3.1) complicates the implementation of the model. Therefore, the model is described as a finite sum of N play operators multiplied with weights which results in:

$$\begin{aligned} y(t) &= \mathcal{P}[u](t) \\ &= p(r_0)u(t) + \sum_{j=1}^{N} p(r_j)F_{r_j}[u](t) \end{aligned} \tag{3.3}$$

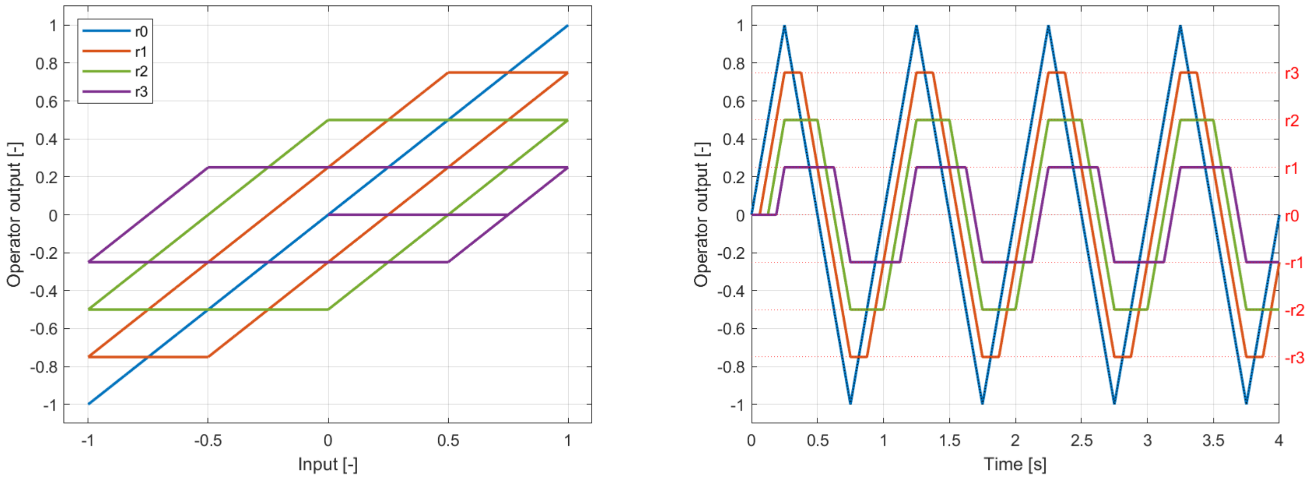
In addition, a term is added to the finite sum in order to compensate for offsets in the model. Figure 3.4b illustrates the PI model with with four thresholds. Here, the effect of the thresholds is visible by the appearances of piecewise elements (which are for this example two times four elements). The smoothness of the hysteresis curve depends on the number of operators defined. The higher amount of operators used, the smoother and more computationally intensive the model gets. Figure 3.5 illustrates the output of each play operator for the unit sawtooth input. In figure 3.5b the boundary stop behavior of each threshold can be observed.



(a) Play operator

(b) Prandtl operator

Figure 3.4: Example of operator output of a (a) single play operator and (b) Prandtl operator with four thresholds when the input is a unit sawtooth.



(a) Input vs. Output

(b) Time

Figure 3.5: Operator output of Prandtl operator with four thresholds when the input is a unit sawtooth when (a) input over output and (b) time where (⁞⁞⁞⁞⁞) is the reference input.

Inverse PI model

The strength of the PI model is that there exist an analytic inverse. The inverse of the PI model is also a hysteresis model and is obtained by the properties of the play operator and its initial loading curve. The following properties hold for the play operator [17]:

- For a given input, **Lipschitz-continuity** of the play operator F_r can be guaranteed for any $r \geq 0$, v(t), and w(t) when

$$|F_r[v](t) - F_r[w](t)| \leq \max_{0 \leq \tau \leq t} |v(\tau) - w(\tau)| \tag{3.4}$$

  satisfies for (3.2c).

- The operator F_r is **rate independent** if

$$F_r[v] \circ \varphi = F_r[v \circ \varphi] \tag{3.5}$$

  where φ is a continuous increasing function $\varphi : [0, T]$ satisfying $\varphi(0) = 0$ and $\varphi(T) = T$.

- The **range of the operator** for a given input $v(t) \in C[0, T]$ and $r \geq 0$

$$\max_{t \in [0,T]} F_r[v](t) = f_r \left(\max_{t \in [0,T]} v(t), w(0) \right) \tag{3.6a}$$

$$\min_{t \in [0,T]} F_r[v](t) = f_r \left(\min_{t \in [0,T]} v(t), w(0) \right) \tag{3.6b}$$

  which indicates that the hysteresis curve is dependent on the input and that it is not possible to model saturation effects.

The concept of the initial loading curve is used to derive the analytical inverse of the model. The initial loading curve is defined when input increases monotonically and the initial state of the model is zero. The initial loading curve describes the hysteresis loop determined by the PI model. The initial loading curve is defined by (3.7) [26].

$$\varphi(r) = p(r_0)r + \int_0^r p(\zeta)g(\zeta)d\zeta \tag{3.7}$$

Taking the first derivative with respect to threshold r gives (3.8).

$$\varphi'(r) = p(r_0) + \int_0^r p(\zeta)d\zeta \tag{3.8}$$

Taking the second derivative with respect to r gives (3.9) which results into weighting function $p(r)$. Furthermore, substituting the zero threshold ($r = 0$) into (3.8) obtains (3.10).

$$\varphi''(r) = p(r) \tag{3.9}$$

$$\varphi'(r) = p(r_0) \tag{3.10}$$

By substituting (3.9) and (3.10) into (3.1) results into (3.11).

$$y(t) = p(r_0)u(t) + \int_0^R \varphi''(r)F_r[u](t)dr \tag{3.11}$$

This result shows that that the PI model can be analytically described using the initial loading curve [27]. Using this result, the finite sum inverse PI model can be described as (3.12) [26].

$$\begin{aligned} u(t) &= \mathcal{P}^{-1}[y](t) \\ &= \hat{p}(\hat{r}_0)y(t) + \sum_{j=1}^{N} \hat{p}(\hat{r}_j)F_{\hat{r}_j}[y](t) \end{aligned} \tag{3.12}$$

where,

$$\hat{r}_0 = 0 \tag{3.13a}$$

$$\hat{p}(\hat{r}_0) = \frac{1}{p(r_0)} \tag{3.13b}$$

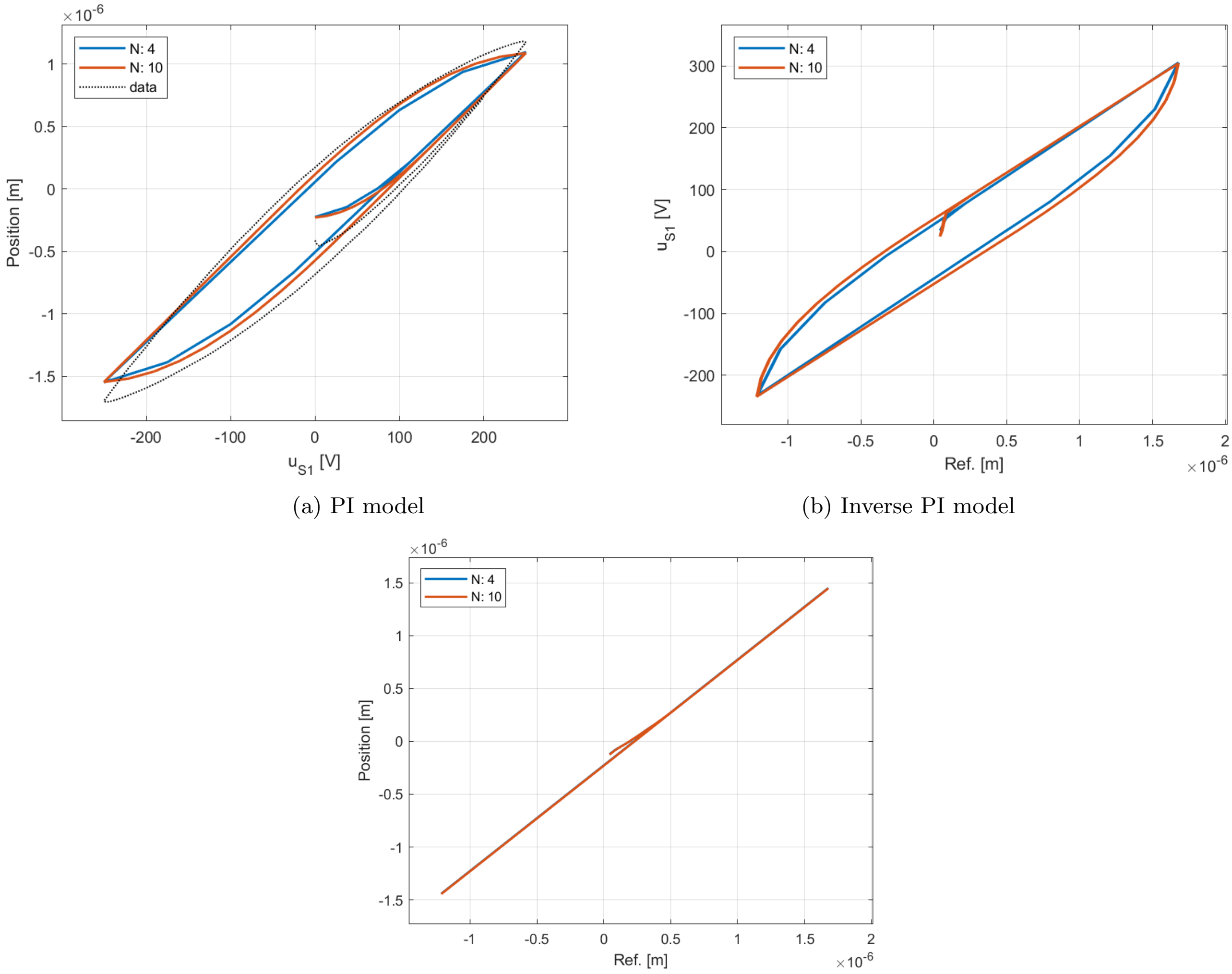
$$\hat{r}_j = p(r_0)r_j + \sum_{i=1}^{j-1} p_i(r_j - r_i), \quad j \geq 1 \tag{3.13c}$$

$$\hat{p}_j = \frac{-p_j}{\left(p_0 + \sum_{i=1}^{j} p_i\right)\left(p_0 + \sum_{i=1}^{j-1} p_i\right)}, \quad j \geq 1 \tag{3.13d}$$

The main idea is to mitigate the hysteresis nonlinearities by using the inverse output of the model as a feedforward compensator since $y(t) = \mathcal{P} \circ \mathcal{P}^{-1}[y](t)$ holds. The first step is to determine the threshold and weight parameters of the forward model. The thresholds are equally distributed over N operators based upon the infinity norm of the input and defined by:

$$r_j = \frac{j}{N}\|u\|_\infty, \quad j = 0, 1, \ldots, N-1 \tag{3.14}$$

Furthermore, r_0 is equal to zero to map the main characteristic of the hysteresis effect. The weights are identified by a least-square fit based upon measurement data of the experimental setup. Next, the computed forward model parameters are used to determine the parameters of the inverse model defined by (3.13). For the implementation, an amount of ten thresholds are used since this gives a good balance between accuracy and computational effort. In addition, an extra lower-order model of four thresholds is defined for comparison. Figure 3.6a, 3.6b, and 3.6c show respectively the simulation outputs of the forward PI model, inverse PI model, and the resulting input vs. output for model orders $N \in \{4, 10\}$. In Figure 3.6a it can be observed that the global characteristics of the hysteresis effect are modeled well. However, due to the symmetric property of the PI model and parameter fit errors of the weights, an increasing model error is visible when approaching the turn-back points (the location where the input direction changes). Furthermore, It can be observed that an increase in operators results in a more accurately and smooth model compared to the lower-order model. Moreover, it can be observed in Figure 3.6c that the concept of a composition function of the forward and inverse models results in a linear system response.



(a) PI model

(b) Inverse PI model

(c) Reference vs. Output

Figure 3.6: Simulation output PI model of model order $N \in \{4, 10\}$ with (a) forward model, (b) inverse model, and (c) composite function $\mathcal{P} \circ \mathcal{P}^{-1}$.

In the next section a generalized envelope function is introduced in the PI model to increase the accuracy and generalization of the model.

3.1.2 Generalized Prandtl-Ishlinskii Model

A generalized operator is used as an extension of the classical PI model that helps to improve the accuracy of the classical PI model compared to the system response. The classical PI model is limited to symmetric hysteresis loops and is unable to show saturation properties. Alternatively, a generalized play operator, which depends on the input, can be utilized by introducing an invertible nonlinear envelope function γ_{lr}. Furthermore, this envelope function is chosen in such a way that the bounds globally describe the increase and decrease curve of the input over the output. In addition, the weights are replaced by a density function p that depends on the thresholds. This extension of the PI model is called a Generalized Prandtl-Ishlinskii model. Figure 3.7a shows the working principle of the generalized play operator. For an asymmetric hysteresis curve, the generalized play operator is constructed out of two envelope functions γ_r and γ_l which respectively represent the increase and decrease curve:

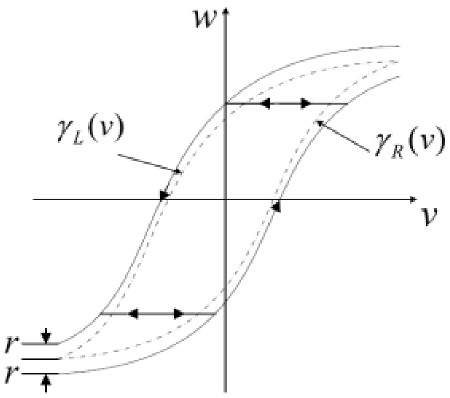
$$F^{\gamma}_{lr}[u](0) = f^{\gamma}_{lr}(u(0), 0) = y(0) \tag{3.15a}$$

$$F^{\gamma}_{lr}[u](t) = f^{\gamma}_{lr}(u(t), F^{\gamma}_{lr}[u](t_i)) \tag{3.15b}$$

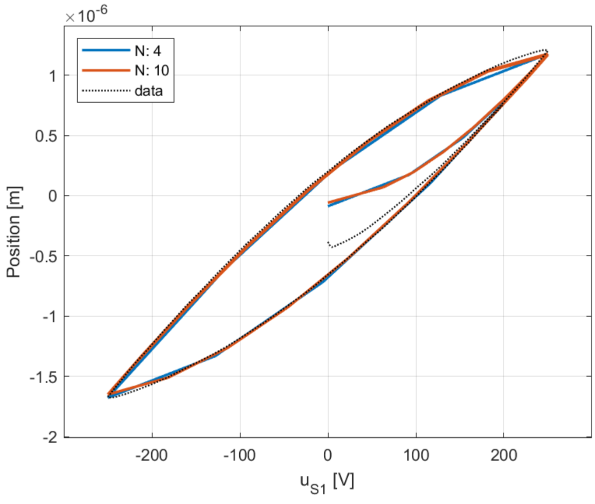
$$f^{\gamma}_{lr}(u, y) = \max(\gamma_l(u) - r, \min(\gamma_r(u) + r, y)) \tag{3.15c}$$

Furthermore, γ_r and γ_l are equal functions for symmetric hysteresis curves. For simplicity and the approximate symmetric hysteresis response of the system, the constructed model is chosen to be symmetric i.e. $\gamma_r = \gamma_l$ with $\gamma_l \leq \gamma_r$. The finite sum GPI model is defined by:

$$\begin{aligned} y(t) &= \mathcal{P}[u](t) \\ &= p(r_0)\gamma(u(t)) + \sum_{j=1}^{N} p(r_j) F^{\gamma}_{lr_i}[u](t) \end{aligned} \tag{3.16}$$



(a) Working principle



(b) Generalized Prandtl operator

Figure 3.7: Working principle of (a) Generalized Prandtl-Ishlinskii hysteresis model with v as input, w as output, and γ_{lr} as the curve function for increasing r and decreasing l [28] and (b) simulation output of GPI model of order $N \in \{4, 10\}$.

Inverse GPI model

As with the classical PI model, the inverse of the GPI model is based upon the relation of the loading curve (3.11). However, the exact equation cannot be applied due to the introduction of the envelope function in the play operator. The following properties hold for the generalized play operator [17][29]:

- For a given input, **Lipschitz-continuity** of the generalized play operator F_r can be guaranteed for any $r \geq 0$, v(t), and w(t) when

$$|F_{lr}^{\gamma}[v](t) - F_{lr}^{\gamma}[w](t)| \leq \max_{0 \leq \tau \leq t} |v(\tau) - w(\tau)| \tag{3.17}$$

  satisfies for the envelope function γ in (3.15c).

- The operator F_{lr}^{γ} is **rate independent** if

$$F_{lr}^{\gamma}[v] \circ \varphi = F_{lr}^{\gamma}[v \circ \varphi] \tag{3.18}$$

  where φ is a continuous increasing function $\varphi : [0, T]$ satisfying $\varphi(0) = 0$ and $\varphi(T) = T$.

- The **range of the operator** for a given input $v(t) \in C[0, T]$ and $r \geq 0$

$$\max_{t \in [0,T]} F_{lr}^{\gamma}[v](t) = f_r \left(\max_{t \in [0,T]} \gamma_r(v(t)), w(0) \right) \tag{3.19a}$$

$$\min_{t \in [0,T]} F_{lr}^{\gamma}[v](t) = f_r \left(\min_{t \in [0,T]} \gamma_l(v(t)), w(0) \right) \tag{3.19b}$$

  which indicates that the hysteresis curve is dependent on the input acting onto the envelope function. Furthermore, saturation and the main curvature of the hysteresis loops are defined by the constructed envelope function.

The finit sum inverse GPI model is described by (3.20) [29].

$$\begin{aligned} u(t) &= \mathcal{P}^{-1}[y](t) \\ &= \gamma_{lr}^{-1} \left(\hat{p}(\hat{r}_0) y(t) + \sum_{j=1}^{N} \hat{p}(\hat{r}_j) F_{\hat{r}_j}[y](t) \right) \end{aligned} \tag{3.20}$$

For the GPI model threshold r is defined by (3.14), envelope function γ, which is based on the linear property of piezoelectric elements, and density function $p(r)$ is defined by:

$$\gamma(u) = c_1 u + c_2 \tag{3.21a}$$

$$p(r) = \rho e^{-\tau r} \tag{3.21b}$$

Furthermore, parameters $c1$, $c2$, ρ, and τ are constants that are identified from experimental data using a least-square fit. Figure 3.7b shows the simulation output of the resulting GPI model for model order $N \in \{4, 10\}$. Here, it is observed that the effect of the envelope and density function increase the accuracy of the model, and explicitly around the turn-back points, compared to the classical PI model.

3.1.3 Prandtl-Ishlinskii MEMS-Model

A disadvantage of the previous PI model methods is that the forward model has to be identified to determine the inverse model. A resulting disadvantage is that the forward model should be accurate enough in mimicking the hysteresis behavior. Model errors are introduced in this process since the model parameters for the PI and GPI model are obtained with an optimizer and therefore depend on a line search algorithm over a defined cost function. In classical feedforward control structures [30], the goal is to obtain an approximate inverse of the system by parameterizing higher-order system behavior like acceleration, jerk, and snap. Usually, the corresponding parameters are found by manually tuning the individual parameters based on output data from closed-loop experiments of the system. This results in an efficient and user-friendly manner compared to optimizers. However, this can be labor-intensive for structures with many parameters.

The goal is to construct an inverse hysteresis model based on the Prandtl-Ishlinskii model that incorporates the benefits of manually parameter tuning. The starting point is to convert the Prandtl-Ishlinskii model to a general memory element (MEMS) as proposed in [31]. The general description of a MEMS-element is given by:

$$y_m(t) = M\left(p_m(t)\right) u_m(t) \tag{3.22}$$

with input $u_m(t)$ and output $y_m(t)$ where M is a one-to-one mapping of the input memory function $p_m(t)$:

$$p_m(t) = p_m\left(t_i\right) + \int_{t_i}^{t} g\left(u_m(\tau)\right) \mathrm{d}\tau \quad \forall t \in \left[t_i, t_{i+1}\right) \tag{3.23}$$

which can be interpreted as momentum with g the input. $p_m(t_i)$ holds the memory value at time instant t_i when the memory function (or momentum) was reset. This is necessary due to the direction changes (turn-back points) in the hysteresis loops. Because of this, $p_m(t_i)$ depends on $p_m(t_i)$ and $u_m(t_i)$ which are described by:

$$p_m\left(t_i\right) = f\left(p_m\left(t_i^-\right), u_m\left(t_i^-\right)\right) \tag{3.24a}$$

$$p_m(t_i^-) = \lim_{s \uparrow t_i} p_m(s) \tag{3.24b}$$

$$u_m(t_i^-) = \lim_{s \uparrow t_i} u_m(s) \tag{3.24c}$$

Next, the Prandtl-Ishlinskii needs to be converted such that it can be described by (3.22). The first step is to cast the play operator as a single MEMS-element. [32] showed that a key property of the hysteresis loop of MEMS-elements is the zero-crossing behavior when the input is zero. In addition, it showed that this property can be achieved by using the first derivative of play operator (3.2c) which results in:

$$\dot{y}(t) = \begin{cases} 0 & \text{if } -r < u(t) - y\left(t_i\right) < r \\ \dot{u}(t) & \text{otherwise} \end{cases} \tag{3.25}$$

This result can be equivalently expressed as:

$$M\left(p_m(t)\right) = \begin{cases} 0 & \text{if } -r < p_m(t) < r \\ 1 & \text{otherwise} \end{cases} \tag{3.26}$$

with reset conditions:

$$f\left(p\left(t_i^-\right), u_m\left(t_i^-\right)\right) = \begin{cases} -r & \text{if } p\left(t_i^-\right) \leq -r \\ p\left(t_i^-\right) & \text{if } -r < p\left(t_{\boldsymbol{i}}^-\right) < r \\ r & \text{if } \quad r \leq p\left(t_{\boldsymbol{i}}^-\right) \end{cases} \tag{3.27}$$

The second step is to cast the Prandtl-Ishlinskii model as a single MEMS-model using the knowledge that the PI model is the weighted sum of N play operators. Consequently constructing the MEMS-model out of N number of elements (equivalent to N number of operators). This results into MEMS-model:

$$M\left(p_m\right) = \sum_{j=0}^{N-1} w_j m_j\left(p_m\right) \tag{3.28}$$

with j-th MEMS-element m_j:

$$m_j\left(p_m\right) = \begin{cases} 0 & \text{if } -r_j < p_{m_j} < r_j \\ 1 & \text{otherwise} \end{cases} \tag{3.29}$$

corresponding j-th weight w_j, reset condition:

$$f_j\left(p_j\left(t_i^-\right)\right) = \begin{cases} -r_j & \text{if } p_j\left(t_i^-\right) \leq -r_j \\ p_j\left(t_i^-\right) & \text{if } -r_j < p_j\left(t_{\boldsymbol{i}}^-\right) < r_j \\ r_j & \text{if } \quad r_j \leq p_j\left(t_{\boldsymbol{i}}^-\right) \end{cases} \tag{3.30}$$

and p_{m_j} the j-th momentum element in momentum vector p_m which is determined from (3.23) with $g(u_m) = [u_m \ \ldots \ u_m]^T$ and $f(p) = [f_0(p_0(t)) \ \ldots \ f_{N-1}(p_{N-1}(t))]^T$.

Inverse PI MEMS-model

The final step is to determine an inverse MEMS-model from the obtained PI MEMS-model. Since M is a one-to-one mapping, a unique computation of an input $u_m(t)$ can be found [32]. Therefore using (3.28) results in inverse:

$$\begin{aligned} \dot{u}_{ff}(t) &= M\left(p_{ff}(t)\right)^{-1} \dot{y}_d(t) \\ &= \frac{1}{\sum_{j=0}^{N-1} w_j m_j\left(p_{ff}(t)\right)} \dot{y}_d(t) \end{aligned} \tag{3.31}$$

with $\dot{u}_{ff}(t)$ the generated input rate, $\dot{y}_d(t)$ the desired output rate (velocity), j-th element weight w_j, j-th MEMS-element m_j:

$$m_j\left(p_{ff}(t)\right) = \begin{cases} 0 & \text{if } -r_j < p_{ff_j}(t) < r_j \\ 1 & \text{otherwise} \end{cases} \tag{3.32}$$

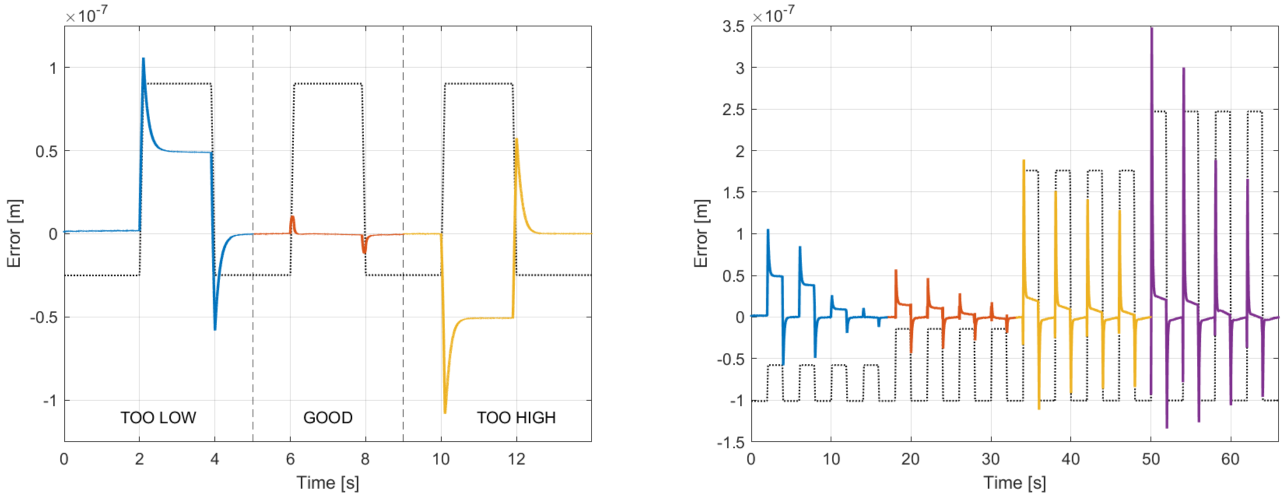
with $p_{ff_j}(t)$ the j-th momentum element of momentum $p_{ff}(t)$ which is determined from:

$$p_{ff}(t) = p_{ff}\left(t_i\right) + \int_{t_i}^{t} \left[\dot{u}_{ff}(\tau) \quad \ldots \quad \dot{u}_{ff}(\tau)\right]^T \, \mathrm{d}\tau \quad \forall t \in [t_i, t_{i+1}) \tag{3.33}$$

with $p_{ff}\left(t_i\right) = f(p_{ff}(t_i^-))$ and $f(p) = [f_0 \ \ldots \ f_{N-1}]^T$ from reset condition:

$$f_j\left(p_{ff_j}\left(t_i^-\right)\right) = \begin{cases} -r_j & \text{if} \quad p_{ff}\left(t_i^-\right) \leq -r \\ p_{ff_j}\left(t_i^-\right) & \text{if} \; -r_j < p_{ff}\left(t_i^-\right) < r_j \\ r_j & \text{if} \quad r_j \leq p_{ff}\left(t_i^-\right) \end{cases} \tag{3.34}$$

Tuning is executed by manually determining the weighting parameter for the corresponding threshold in order from r_0 to r_{N-1}. Here, the piezo element is actuated with a step input that has an amplitude between the current and next threshold value. Therefore, the amplitude increases when tuning progresses. As with classical feedforward tuning, the resulting error output is used to tune the weight of the activated threshold. The goal is to reduce the error to zero by increasing or decreasing the weighting value such that only non-mappable higher-order errors remain. Figure 3.8a illustrates the error output for three tuning cases that occur. In the first case, when the parameter value is set too low (▬), the error is positive. In the second case, when the parameter value is set too high (▬), the error is negative. In the last case, when the parameter value is set correctly (▬), the error should be zero. The full procedure is explained in [31]. For simplicity of implementation effort, the number of elements N is set as 4. The corresponding thresholds r are defined by (3.14). Figure 3.8b shows the error output while tuning the parameters of the defined model. A step signal (┈) with an increasing amplitude is applied to the shear element to activate the corresponding threshold. Tuning of the weighting parameters w_0, w_1, w_2, w_3 is illustrated by (▬), (▬), (▬), and (▬) respectively.



(a) Method

(b) Process

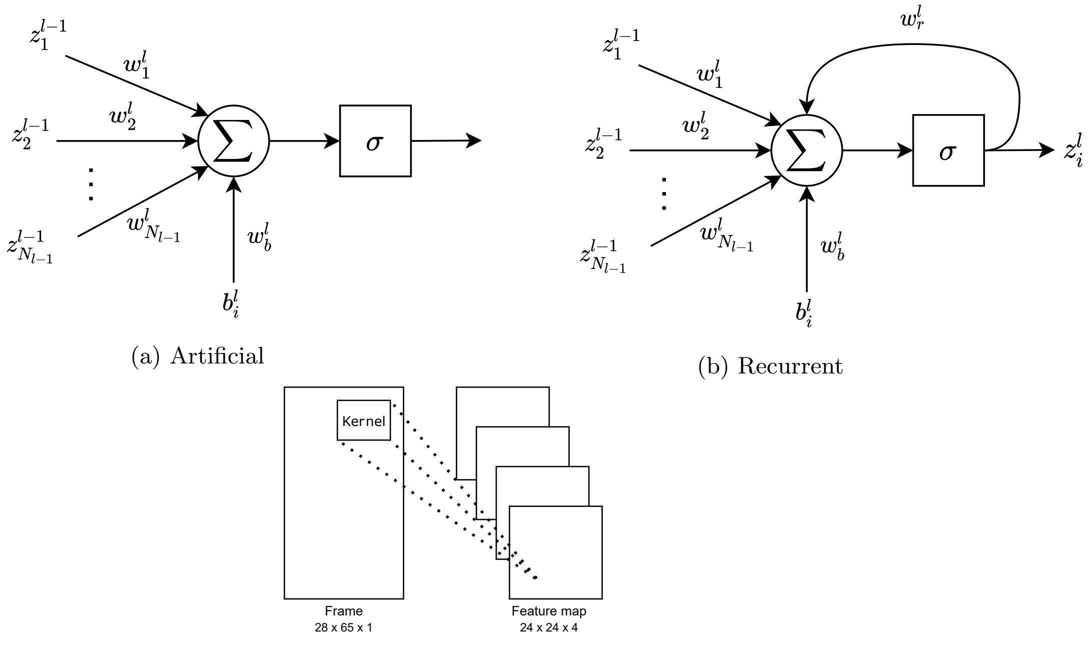
Figure 3.8: (a) MEMS tuning method for tuning of weight parameters w_j with the error output when the parameter value is (▬) too low, (▬) correct, and (▬) too high. (b) The error output while parameter tuning of the proposed MEMS model with $N = 4$ elements where (┈) is the scaled reference input, (▬) is the tuning of parameter w_0 in series up to parameter (▬) w_3.

3.1.4 Prandtl-Ishlinskii Neural Network Model

A Neural Network is a function approximator that is able to predict/map an outcome depending on input data applied. Common applications of neural networks are in the classification of images [33], character recognition [34], speech recognition [35], and adaptive control for automotive

[36]. Besides classification and recognition tasks, neural networks are also capable of predicting trajectories on time series data such as stock prices [37] and system behavior [38]. The high-level structure of a neural network consists out of an input layer, hidden layer, and an output layer. The input layer accepts the input features of the data flow in the system and forwards these to the hidden layer. The hidden layer consists out of a collection of layers which task is to extract and manipulate the input data. The output layer combines the outcome of the hidden layer and predicts/maps this to a value.

Three common neural network structures are Convolution Neural Network (CNN) , Artificial Neural Network (ANN), and Recurrent Neural Network (RNN). A CNN is a network structure that uses kernels (filters) in combination with a convolution operation to extract features within data. Common applications of CNNs are in classification and recognition for images, video, and audio. An ANN is a network structure that is based on the neural connections inside the human brain. Here, each layer consists out of collections of neuron nodes as illustrated in Figure 3.9a and 3.9b. An RNN, and its successors Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), is a more advanced type of ANN that can handle dependencies within the data due to the addition of an internal feedback loop. Therefore, RNNs are commonly used in applications where dependency such as past inputs and outputs relates. Application examples are in machine translation, speech recognition, time series prediction, and human action recognition. Moreover, as an effect of the internal feedback loop, an increase of instability (due to an exploding gradient [39]) and computational effort is obtained during the training of the parameters.



(a) Artificial

(b) Recurrent

(c) Convolution

Figure 3.9: Neural network node representation for (a) Artificial Neural Network, (b) Recurrent Neural Network, and (c) Convolution Neural Network.

Nodes can either have single or multiple inputs and outputs. Furthermore, layers are interconnected in such a way that the output of a node forms the input for the next layer. The value of a node is determined by the weighted summation of the inputs of that particular node. In addition, a bias can be added to the summation such that the node gains a higher interest compared to other nodes. A single ANN node is defined by:

$$z_i^{(l)} = \sigma_i^{(l)} \left(\sum_{j=1}^{N_{l-1}} w_j^{(l)} z_j^{(l-1)} + b_i^{(l)} \right) \tag{3.35}$$

with $z_i^{(l)}$ the value of the i-th node in layer l, $w_j^{(l)}$ the weight for the j-th input, $b_i^{(l)}$ the bias value, $\sigma_i^{(l)}$ the activation function, and N_{l-1} the number of outputs of the previous layer. The values of a node are inserted into an activation function [40] such that the node outputs only relevant information to the next layer. Therefore the choice of activation function determines the specification of relevant information. In addition, activation functions help with normalizing the output such that computational effort decreases when node values increase [41]. The choice of an activation function determines how the network can handle the data and therefore the prediction. Activation functions are often non-linear and differentiable functions. Non-linearity allows the network to train non-linear relationships in the data. The differentiability of an activation function allows backpropagation [42] of the model's error when training to optimize the weights.

Parameters of the network such as weights and biases are trained to get an accurate prediction of the network. The parameters are trained with input and output data from the system that is mimicked. A loss function and optimizer are defined for the training process. The loss function determines the performance of the network with the trained parameters compared to the target data. If the output of the loss function is big compared to the performance value, then the system does not perform well and needs to be retrained. The optimizer defines the optimizing function used for parameter training. Here, the aim is to minimize the cost function by adjusting the network parameters. Furthermore, in backpropagation the parameters are determined by a backward pass through the network with the gradient of the last layer of weights calculated first and the gradient of the first layer of weights calculated last. Most used optimizers are based on the gradient descent method [43]:

$$\theta_{j+1} = \theta_j - \alpha \nabla f(\theta) \tag{3.36}$$

Here, the new parameter values are determined by the current values θ_j, an update rate α, and the gradient of a cost function $\nabla f(\theta)$ based on the parameters. The step size of the updated values is determined by the gradients of the cost function and the update rate. The update rate plays an important role during training. If the update rate is too big, a local minimum could not be reached due to a resulting bouncing behavior within the cost function. Furthermore, if the value is too small, gradient descent will settle to the local minimum in a long period of time.

During backpropagation, the gradients are either amplified or minimized when going from the last layer to the first. This is since backpropagation is the traversing chain rule between layers, i.e., continuous matrix multiplications. For large derivatives, the gradient will increase exponentially when propagating further in the model resulting in a so-called exploding gradient problem. This leads to too much variation in updated weights. Alternatively, for small gradients, it will decrease exponentially and vanishing resulting in a vanishing gradient problem. This leads to a very small change in the updated weights. In both cases the parameters will not converge and is therefore unwanted. To prevent these two phenomena, two properties need to hold. The first property is that the mean of the activations (value of nodes) should be zero. The second property is that the variance of the activations should be equal in each layer. To meet these properties all the initial weights are picked randomly from a normal distribution with zero-mean and complying variance.

Training is realized by iterations of forward and backpropagation actions for a defined data set. Each iteration is depicted as an epoch which is defined as the completion of a full dataset. In a single iteration, the loss function values of the network for the train and validation data set are computed. Based on this training loss, backpropagation is performed. After backpropagation is finished a new iteration with the updated parameters starts. Training ends when an exit condition is met. Usually, these conditions are defined as the maximum amount of epochs, achieved training loss value, or as the validation loss increases for a defined amount of epochs.

Neural Networks for hysteresis modeling

In literature, neural networks for hysteresis modeling are used in a variety of methods. For example, in [44] a rate-dependent PI model is combined via a dot product operation with a convolution neural network. The rate-dependent model maps the rate-dependent hysteresis behavior of the stepper by the use of the Fourier transform of the input signal. The convolutional network extracts the deep features of the input signal to improve the generalization ability of the model. This extends the rate-dependent PI model and results in an increase in accuracy compared to the PI model. Moreover, in [45] a recurrent neural network is proposed to predict the hysteresis behavior based upon the Preisach play operator. Here, the output of the Preisach play operator in combination with the position and velocity trajectory is used as the input of the network. Furthermore, this research showed that the addition of the play operator increased performance by 19% for the major loop and 44% for the minor loop hysteresis compared to the trained network without the operators.

Forward PI-NN model

A Prandtl operator-based neural network (PI-NN) is constructed to predict the hysteresis phenomena. The network is defined in the Python programming language with the use of the PyTorch [46] neural network framework. PyTorch is an open-source framework with high-level support of neural network building blocks such as basic and complex network structures, activation functions, and tensor manipulation. An Artificial Neural Network structure is used because the network itself does not have to memorize any past dynamics of the system. This is due to the history property of the play operator which in this case is connected to the input of the network and therefore transfers the dependency of the past dynamics into the network. The forward network, illustrated by Figure 3.10, consists out of N inputs that are depending on the number of play operators, 4 hidden layers with decreasing node sizes starting at $4N$, and a single output. All hidden layers are connected in series with a Rectified Linear Unit (ReLU) activation function [40]. A ReLU function performs better network generalization compared to the tanh and sigmoid functions [47]. In addition, due to its nearly linear behavior, the computational effort is low compared to the tanh and sigmoid functions. Furthermore, the function forces values that are less than zero to zero and therefore eliminates the vanishing gradient problem. Input data of the Prandtl operators are normalized between 0 and 1 to decrease computational effort. The output data of the network is re-scaled based on the normalization factors. Normalization factors k_u and k_y are defined so that they span the system boundaries of the input voltage and the output displacement.

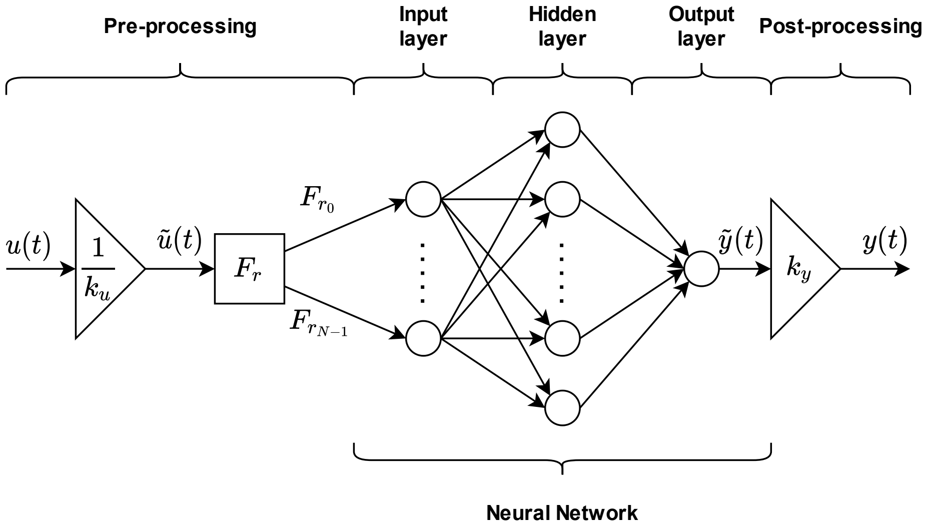
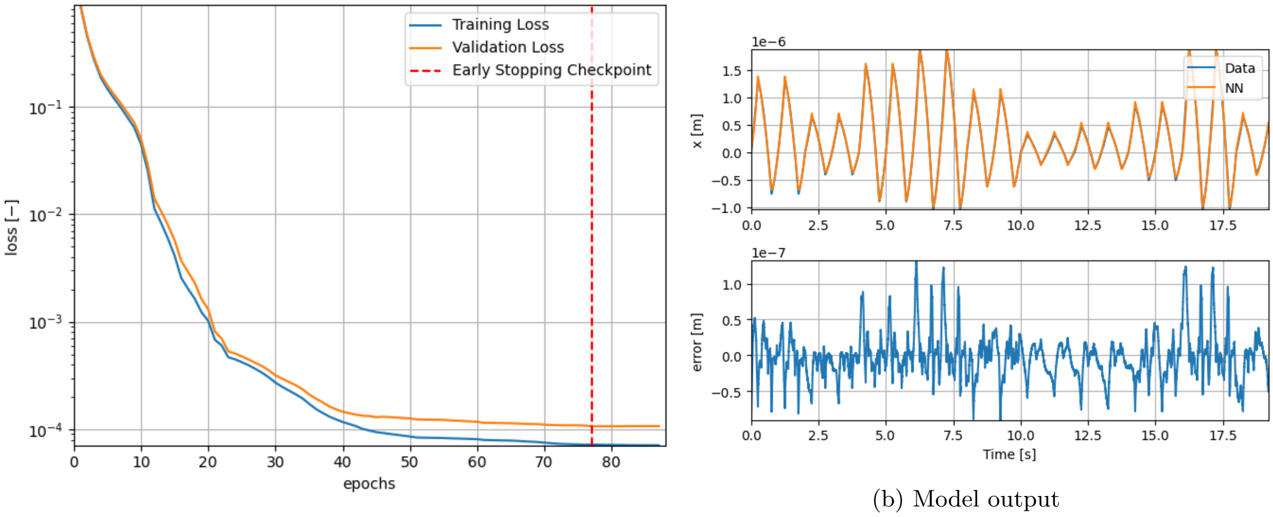


Figure 3.10: Block diagram of artificial Prandtl-Ishlinskii Neural Network model.

The proposed network is trained with different amounts of Prandtl operators to investigate the effect of the operators on the network. For parameter training, the loss function is defined as the Mean Square Error (MSE) between the prediction and the training data. The optimizing algorithm is the Broyden–Fletcher–Goldfarb–Shanno (LBFGS) algorithm [48]. The LBFGS algorithm is a extension of the Quasi-Newton method and makes use of the Hessian matrix and an approximation of its inverse. As a result that less memory is needed for big datasets and parameter determining is achieved faster and more accurate compared to other standard optimization algorithms within PyTorch. Furthermore, the training data is constructed out of measurement data from the experimental setup, and the thresholds are defined by (3.14). The measurement data is split into a train and validation set that respectively represent 70% and 30% of the full data set. Initialization of the weights is performed by the Glorot algorithm as proposed in [49]. Networks of multiple operator amounts are defined to analyze the effect of the Prandtl operator. Table 3.1 shows the training output of the forward network for different operator amounts. It can be observed that the number of operators affects the time needed to train the network. This is expected since the number of nodes and connections in the network increase rapidly since the first hidden layer consists out of $4N$ nodes. Furthermore, a decreased error is observed for increasing operator models. Figure 3.11a shows the training and validation loss for the network model with 10 operators. It can be noticed that the loss functions converge and that the training has been stopped by the increasing validation loss criterion which was activated at epoch 77. Furthermore, Figure 3.11b shows the model output for the 10-operator model with the best parameter fit i.e. lowest loss value before the validation loss increases. Therefore, concluding that the neural network is capable of mimicking the hysteresis behavior. The next step is to construct an inverse model such that this can be used for feedforward control.

Table 3.1: Training output of forward Prandtl-Ishlinskii neural network for multiple operator numbers.

#		Training		Prediction	
Operators	Train time	$\vert\vert e\vert\vert_2$	$\vert\vert e\vert\vert_\infty$	$\vert\vert e\vert\vert_2$	$\vert\vert e\vert\vert_\infty$
	[s]	[-]	[μm]	[μm]	[-]
5	63	11.692	0.119	16.437	0.181
10	153	8.648	0.124	10.475	0.154
15	289	7.104	0.0981	12.716	0.141



(a) Cost function loss

(b) Model output

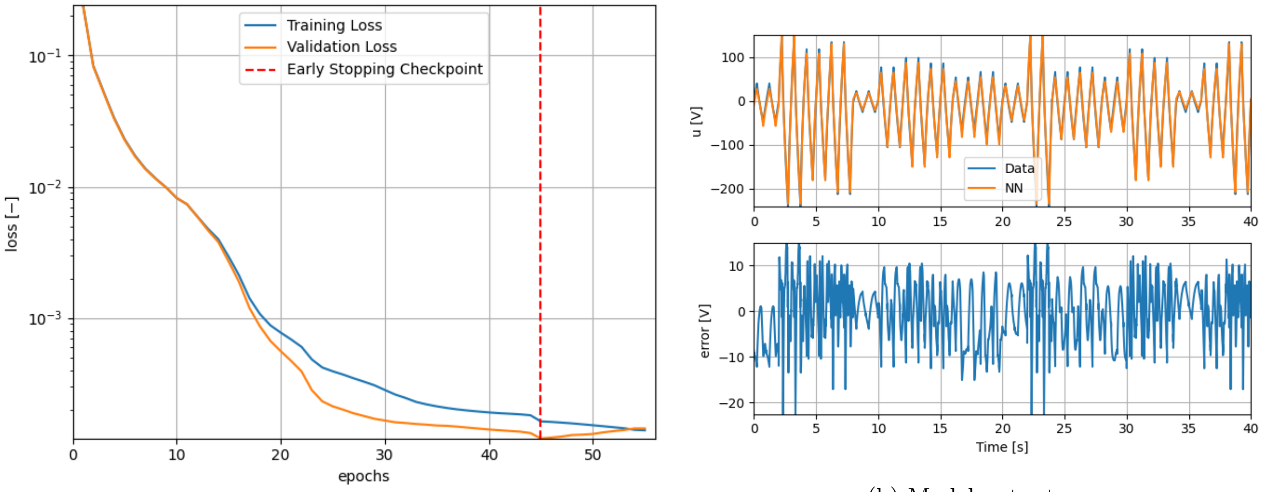
Figure 3.11: Forward neural network training output from network with 10 operators where (a) loss function of training and validation datasets and (b) model output of best trained model.

Inverse PI-NN model

An inverse PI neural network is constructed for hysteresis compensation. Since a neural network only maps a function, a direct inverse such as with the classical PI model is not possible. For simplicity, the inverse model is constructed as an approximate model based on experimental data from a mimicking situation and inverse GPI parameters. The target data is derived by the output signal of an iterative learning controller applied to the shear elements in open-loop. The inverse GPI parameters are duplicated from the model obtained in Section 3.1.2. The network structure is the same as for the forward network. Table 3.2 shows the training output of the inverse network for different operator amounts. Due to cross-compatibility issues between the PyTorch neural network structure and the Simulink control structure, it is not possible to implement the neural network model structure-based. Therefore, the output sequence of the inverse PI-NN model with the defined reference input is converted such that it can be implemented as a repeating sequence within the control structure. For this reason, it is not possible to do experiments with a different reference input.

Table 3.2: Training output of inverse Prandtl-Ishlinskii neural network for multiple operator numbers.

#		Training		Prediction	
Operators	Train time	$\|\|e\|\|_2$	$\|\|e\|\|_\infty$	$\|\|e\|\|_2$	$\|\|e\|\|_\infty$
	[s]	[-]	[V]	[-]	[V]
5	155	3443.413	25.203	1199.738	24.837
10	230	1871.734	17.661	629.908	10.772
15	524	2047.675	24.152	1061.854	20.452



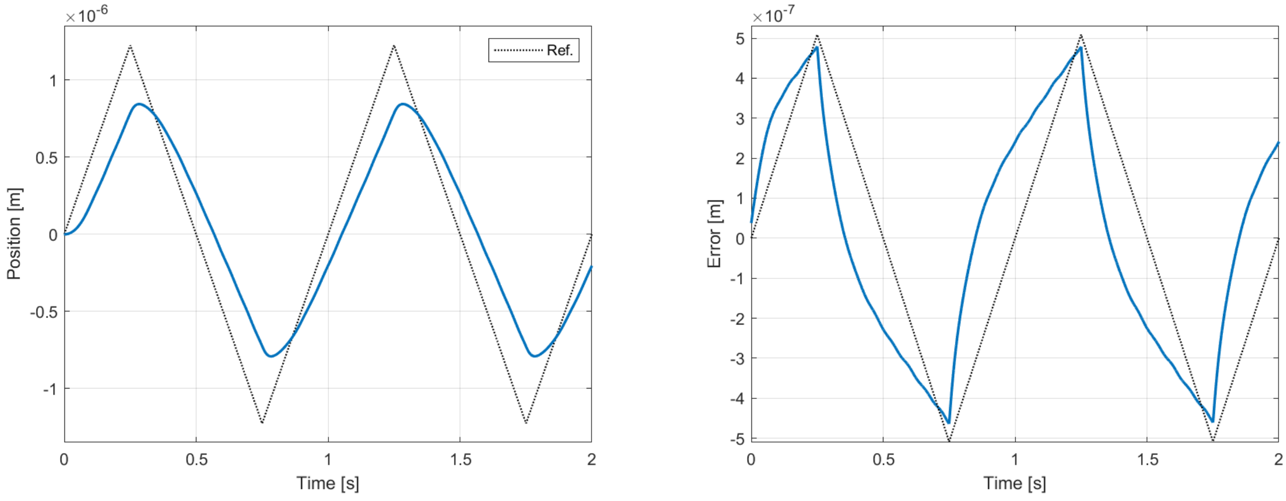
(a) Cost function loss

(b) Model output

Figure 3.12: Inverse neural network training output from network with 10 operators where (a) loss function of training and validation datasets and (b) model output of best trained model.

3.2 Hystersis Compensation by the Prandtl-Ishlinskii Models

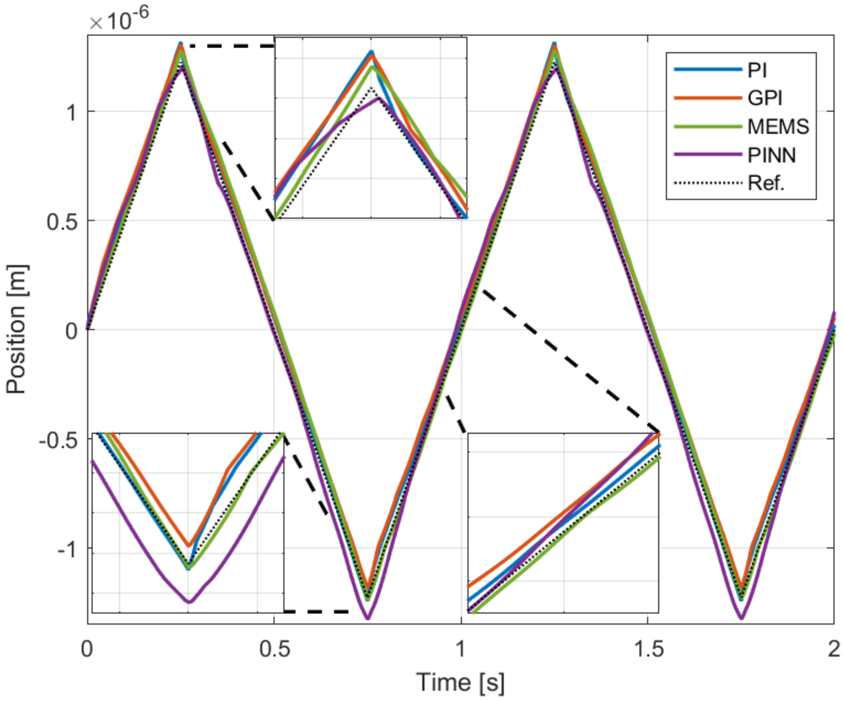
The inverse hysteresis models are implemented and tested in the closed-loop environment. Here, the models are implemented as a feedforward controller as discussed in Section 2.1.1 and illustrated in Figure 2.3b. Moreover, the input reference is defined as a 1 [Hz] sawtooth signal with an amplitude of half an actuator step. The resulting uncompensated closed-loop response, shown in Figure 3.13, illustrates the effect of hysteresis on the system's performance which results into a peak error of approximately 0.5 [μm] with the defined reference input.



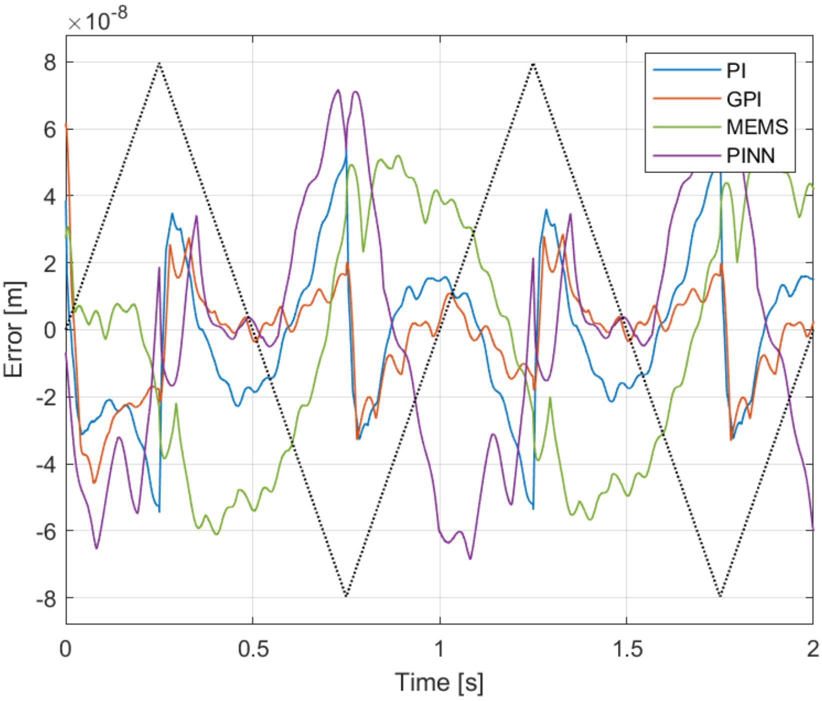
(a) Position (b) Error

Figure 3.13: System output while walking with no hysteresis compensation with (a) the position of the mover with (⋯⋯) the reference input and (b) the error with (⋯⋯) the scaled reference input.

For this research, the classical Prandtl-Ishlinskii model and its variations are implemented as a feedforward compensator. Furthermore, the number of operators for the best model comparison is chosen as ten since this resulted in the best output compared to the four operator case (see Table 3.3 for the error properties). Note, the MEMS-model constructed has only four elements. The resulting system output while compensated by these models is shown in Figure 3.14a. Here, it can be observed that the models can improve system accuracy by compensating for the hysteresis behavior which results in an increase of the reference tracking compared to the uncompensated situation. This accuracy is determined by the correctness of the parameter fit of the model. When observing the zoomed-in figures, it shows the inaccuracies that occur due to the modeling errors. Particularly, this is visible at the regions where the input rate changes direction (turn-back points at $t = \{0.25, 0.75, 1.25, 1.75\}$ [s]). The resulting error after compensation is shown in Figure 3.14b. The mean of the error output has been used since the initial loading curve of the model results in an offset in contrast with the reference input. Overall the models tends to have a comparable result. However, a difference can be observed for the error curve of the MEMS model. Here, the curve is resulting in a more sinusoidal error curve compared to the other models. Furthermore, it can be observed that all the models lose accuracy at the regions where the input rate changes direction as also observed in the zoomed-in figure of the position output. The RMS error, infinity norm, and peak-to-peak difference of the error are displayed in Table 3.3. The resulting values confirm that the considered hysteresis compensators decrease the effect of hysteresis and therefore improve system response. In addition, the values indicate that the models result in comparable system output. When comparing the error values of each model it shows that the GPI model performs best out of the implemented models. Furthermore, it can be concluded that the PI-NN model performs the worst. This is a result of not being a direct inverse of the forward model and therefore have a higher degree of model errors. This can also be seen in the turn-back points in Figure 3.14a. Contrary to all models, the MEMS model discussed in this report only uses four elements. However, the performance is by approximation similar to the other models. Therefore, the performance would increase and outperform the other models when equalizing the number of elements in the model. Table 3.4 gives an overview of the pros and cons of each modeling method.



(a) Position



(b) Error

Figure 3.14: Hysteresis compensation output for all hysteresis models with (a) the position fo the mover and (b) the centered error with (⋯⋯) scaled reference input.

Table 3.3: Error properties of hysteresis compensator models.

Compensator model	Operators	RMSE [nm]	$\|\|e\|\|_\infty$ [nm]	e_{PP} [nm]
None	-	286.246	482.872	946.993
PI	4	26.114	62.144	120.406
PI	10	22.059	55.475	109.3638
GPI	4	12.286	70.320	119.448
GPI	10	12.481	61.046	107.699
MEMS	4	36.515	61.149	117.201
PI-NN	4	43.425	120.295	199.798
PI-NN	10	37.863	71.262	140.978

Table 3.4: Pros and cons of used model methods.

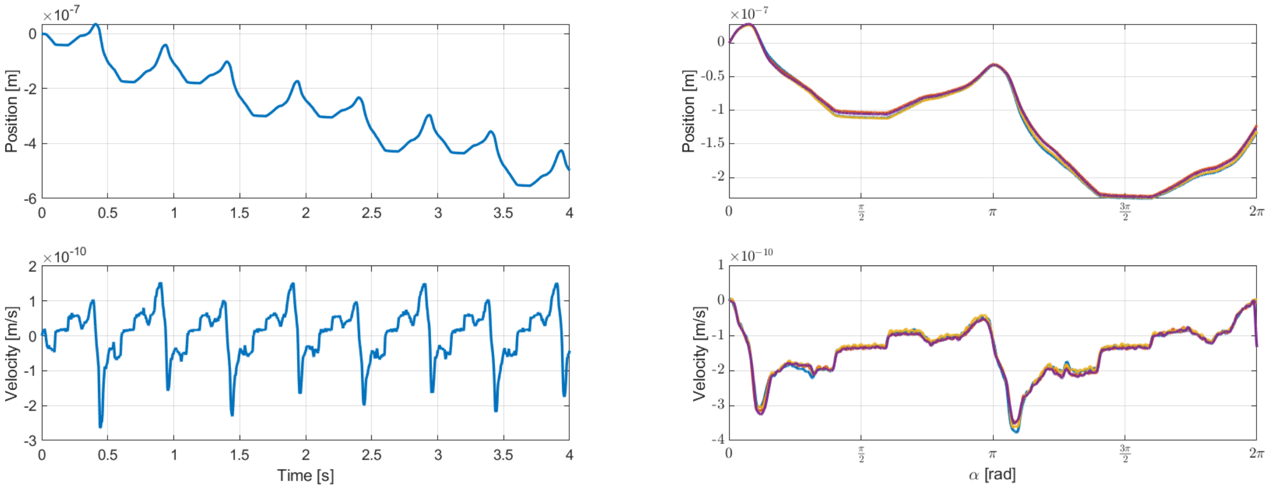
Model	Advantage	Disadvantage
PI	• Simple model • Analytical inverse	• Parameter fitting ↪ Accuracy decrease by less advanced optimizer and possible local minima. • Basic hysteresis curve ↪ No saturation ↪ No asymmetric loops
GPI	• Simple model • Analytical inverse • Hysteresis curve shaping by envelope function. ↪ Saturation possible ↪ Asymmetric loops possible	• Parameter fitting ↪ Accuracy decrease by less advanced optimizer and possible local minima.
MEMS	• Analytical inverse • Simple feedforward tuning ↪ Manually ↪ Linkable to physical component • High performance with few elements	• Advanced implementation • Labor intensive for higher number of elements
PI-NN	• Generic approach of modeling	• Advanced structure ↪ Abstract internals • No analytical inverse • Possible implementation incompatibility • Accurate training data needed

Chapter 4

Control of a Hysteric Piezo Steppers

Piezo steppers are well known for their implementation in nanopositioning systems. To comply with the high-precision requirements, the actuator is controlled by a high-end advanced precision controller. Most mechatronic systems use a traditional controller such as feedback or feedforward controllers. In the case of a feedback controller, the difference between the output and the reference is used to adjust the system correctly. An advantage of a feedback controller is that disturbances and noisy signals can be rejected and its structure type is simple. A disadvantage is that the controller takes corrective actions after the actuation of the system. A feedforward controller does not make use of the error between the reference and the output but makes use of prior knowledge of the system. Here, higher-order signals like acceleration, jerk, and snap, are used as a tool to generate prior known errors for the system. Moreover, the general idea behind a feedforward controller is to determine an approximation of the inverse of the real system. As a result, in an ideal case, the reference is the same as the output. However, this is not the case in a real environment due to inaccuracies in the approximation. An advantage of feedforward is that the system does not react based on the previous response. A disadvantage is that the controller variables are tuned based on pre-obtained experimental data. Both the traditional form of the controller types are fixed and are not able to correct themselves (i.e. update controller parameters) when task or reference is changing.

Much research is dedicated to extending and improving the feedback and feedforward structures. For example, iterative learning control [50] have shown that for systems with repeating tasks a significant performance increase can be achieved compared to traditional feedback controllers. For this reason, the system is investigated for structures of repetitive behavior. Figure 4.1a shows the position and velocity response of the experimental setup when only clamp set 1 is actuated with a saw-tooth input of 1 [Hz] for 4 seconds i.e. 4 actuation waves. In the tp figure it can be seen that there is some repetitive behavior with an additional trend. Furthermore, the velocity figure confirms that there are repetitive structures in the setup. However, since there is a trend and the actuation periods are frequency depended, the time domain data could not be thoroughly used for iterative/repetitive controller types. Figure 4.1b shows the same system response but then in the commutation angle domain. By using the same data in the commutation angle domain a clear repetitive system behavior can be observed. Here, no trends are visible when comparing each wave set to each other. In addition, other experiment results showed that for other step frequencies similar repetitive structures hold. Therefore, the commutation angle domain could be used for iterative control of the system. For this reason, the implementation of an iterative learning controller for the experimental setup in walking behavior is further discussed in this chapter.



(a) Time domain

(b) Commutation angle domain

Figure 4.1: System response when only clamp set 1 is actuated with 1 [Hz] for 4 seconds in (a) time domain and (b) commutation angle domain.

4.1 Iterative learning control for Piezo steppers

Iterative Learning Control (ILC) is a common method of learning control that makes use of the past error data to improve control in future trials (previous experiments). Here, the error of the previous trial is used to update the controller for the current trial. An important condition within iterative control is that the trials consist out of the same task. Therefore, each task should always start (initialized) at the same point as the previous. In case of repetitive tasks without initializing, a repetitive learning controller [51] should be used instead of an iterative controller. Common applications for ILC are in welding processes [52], industrial printer systems [53], or wire bonding [54]. An advantage of ILC is that due to repetitive behavior it can handle non-causal control schemes. However, a disadvantage is that the controller cannot deal efficiently with changing reference signals. As a result, the error increases and the system needs to re-tune its parameters. In literature, a variety of methods are available for implementation with increased flexibility and performance. Examples are frequency ILC [55], lifted ILC [56], increasing reference flexibility by the introduction of ILC with basis functions [57], and projection-based ILC [58] where non-repetitive error components are extracted by projecting these to a subspace.

The basic framework for ILC is either frequency ILC, which makes use of the frequency domain response of the system, or lifted ILC, which makes use of the impulse response of the system. Within both frameworks learning filter L and robustness filter Q are applied to respectively achieve stability and robustness of the system. Figure 4.2 illustrates the lifted ILC framework with J being a Toeplitz matrix composed of Markov parameters of the system response. The determination of the Q and L filter depends on the ILC method used. Usually, for the frequency domain, L is obtained by the inverse of the process sensitivity and Q by a filter that is 1 for frequencies where

$1 - JL < 1$, and $|Q| < 1$ such that convergence criteria $|Q(1 - JL)| < 1$ holds for all ω. Generally, for the lifted method, L and Q are obtained by an optimal synthesis in the form of a quadratic cost function. The cost function is minimized over three terms: error, input, and the input rate. Additionally, weights are multiplied to their terms to change the individual contribution within the function.

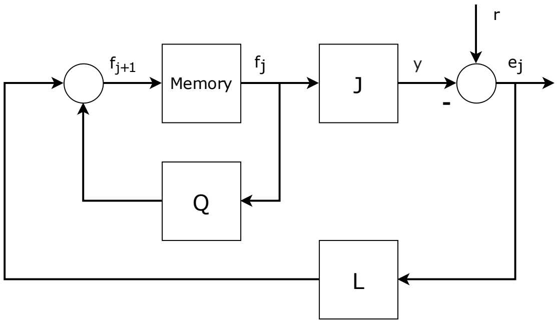


Figure 4.2: Block scheme of a lifted ILC setup with system J in sensitivity feedback form, learning filter L, robustness filter Q, input f_j, output y, reference r, tracking error e_j, and next input f_{j+1}. The trial index is denoted as j.

For application-specific behavior, the system should be able to deal with step frequency varying tasks during walking behavior. A consequence of step frequency varying tasks is that the number of samples within the commutation angle domain varies. This result into a deterioration of the control signal which is shown in Figure 4.3. In addition, a change in the reference signal results in an increased error after the transition. ILC showed that it is capable of rejection of disturbances that repeat during tasks. However, performance and sampling degrades when varying disturbances are introduced in each iteration. By adding basis functions to ILC, system responses such as the error are parameterized to obtain continuous description of that signal. Therefore, ILC with basis functions is introduced to the system. In addition, the control structure should be able to compensate for the hysteresis phenomena with the implementation of the Generalized Prandtl-Ishlinskii compensator from Section 3.1.2. The following sections discuss the theory of ILC with basis functions, the choice of the basis function sets used in this research and the implementation method with and without hysteresis compensation.

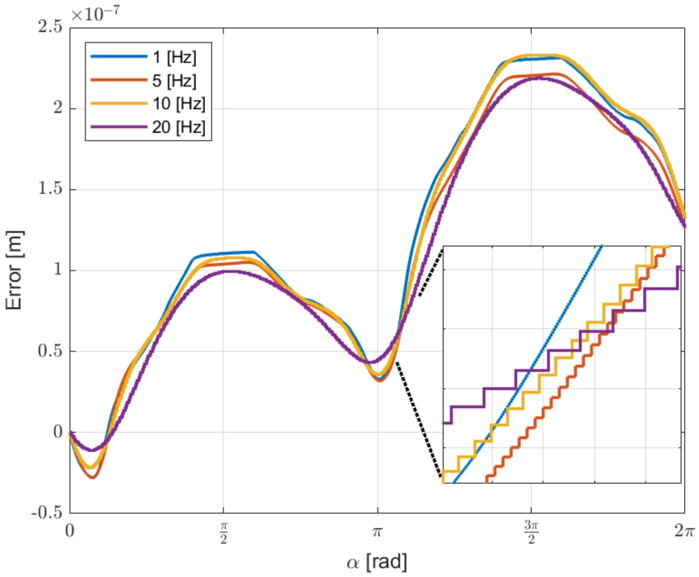


Figure 4.3: Signal deterioration due to varying sample sizes in the commutation angle domain for varying step frequencies.

4.1.1 Iterative Learning Control with Basis Functions

Iterative learning control with basis functions compensates disturbance acting on the system by making use of pre-defined function sets that model the characteristics of the disturbance. Here, partial error information from a previous iteration in the commutation angle domain is obtained by projecting the entire error onto a smaller subspace spanned by a set of basis functions. This projection is defined by:

$$f_j(\alpha) = \psi(\alpha)\theta_j \tag{4.1}$$

and consist out of a set of linear independent functions $\psi(\alpha)$:

$$\psi(\alpha) = [\psi_1(\alpha)\ \psi_2(\alpha)\ \ \dots\ \ \psi_M(\alpha)] \in \mathbb{R}^{1\times N} \tag{4.2}$$

and a vector of corresponding weighting parameters θ:

$$\theta_j = [\theta_{1,j}\ \theta_{2,j}\ \ \dots\ \ \theta_{M,j}\ ]^\top \in \mathbb{R}^{M\times 1} \tag{4.3}$$

The control structure as discussed in Section 2.1.1 is illustrated by Figure 2.3c. The system relation is defined by:

$$y_j = SGCr + SGf_j + Sv_j \tag{4.4a}$$

$$e_j = r - y_j \tag{4.4b}$$

with G the system, $S = (I+GC)^{-1}$ the sensitivity function, $F(\theta)$ the basis function with output f_j [V], C the feedback controller with output u_j^c [V], r [m] the reference, e_j [m] is the reference tracking error, y_j is the system output, and v_j [m] the disturbance acting on the system. When substituting (4.4a) in (4.4b), the error results into $e_j = [I - GF(\theta_j)]Sr - Sv_j$. Optimal reference tracking, i.e. $e_j = -Sv_j$ is achieved when controller model $F(\theta) = G^{-1}$. However, this is not possible in practice since only approximate models of the inverse plant are available. Therefore, resulting into deterioration of performance. The update law for the disturbance compensator is constructed using the basis function prediction of the disturbance, a learning gain L and the last compensator input:

$$u_{j+1}(\alpha) = u_j + L\psi(\alpha)\theta_{j+1} \tag{4.5}$$

The optimal update values of parameter vector θ_j is obtained by a least square fit of the difference between the error and the basis function prediction:

$$\mathcal{J}(\theta_j) = \sum_{i=1}^{N_j} \left(e_j(i) - \psi\left(\alpha_j(i)\right)\theta_j\right)^2 \tag{4.6}$$

To find the optimal θ_j with a minimal $\mathcal{J}(\theta_j)$

$$\frac{\partial \mathcal{J}(\theta_j)}{\partial \theta_j} = 0 \tag{4.7}$$

needs to hold. Using the first order necessary condition for optimality, optimal θ_j is determined by:

$$\begin{aligned} \theta_{j+1} &= \left(\psi^\top \psi\right)^{-1} \psi e_j \\ &= \psi^\dagger e_j \end{aligned} \tag{4.8}$$

Here, the pseudo inverse of the basis function is multiplied with the error. Furthermore, $\psi^\top\psi$ represents the Gramian matrix of the basis functions. It is necessary that the individual basis function vectors ψ_j are linear independent in order to obtain a positive definite matrix property and therefore obtain invertibility of the matrix. Only then an unique point that minimizes the convex cost function can be found. When substituting optimal θ_j (4.8) into (4.5) we find the final update law for the next iteration.

The system relationship in lifted notation is defined by:

$$y_j = c(u_j + v_j) \tag{4.9a}$$

$$e_j = r - y_j \tag{4.9b}$$

When substituting lifted output relation (4.9a), lifted learning equation (4.5), and the projection equation (4.8) into the next trial error (e_{j+1}) (4.9b), the error evolution is obtained:

$$\begin{aligned} e_{j+1} &= r - y_{j+1} \\ &= e_j - cL\psi\theta_j \\ e_{j+1} &= \left(I - cL\right) e_j \end{aligned} \tag{4.10}$$

where I is identity and c is the piezo constant [mV^{-1}]. Therefore for a error decrease, learning gain L should be chosen such that $|I - cL| < 1$ holds.

4.2 Construction and Selection of the Basis Function

The set of basis functions are chosen based on prior knowledge of the system's behavior like disturbances. For this application, there are two types of sets:

$$\psi(\alpha) = [\psi_{S1}(\alpha)\ \psi_{S2}(\alpha)\ \psi_c(\alpha)\ \psi_s(\alpha)]^\top \tag{4.11}$$

$$\theta_j = [\theta_{S1,j}\ \theta_{S2,j}\ \theta_{c,j}\ \theta_{s,j}]^\top \tag{4.12}$$

The first set, functions $\psi_{S1}(\alpha)$ and $\psi_{S2}(\alpha)$, are based on the input waveforms for the individual shear elements $S1$ and $S2$. Here, corresponding parameters θ_{S1} and θ_{S2} representing the scaling factor for the shear waveform. Because of this, the shear steps size can be increased or reduced in case the step rate is respectively too low or too high compared to the reference. The second set, functions $\psi_c(\alpha)$ and $\psi_s(\alpha)$:

$$\psi_c(\alpha) = \left[\cos\left(2\pi\frac{\alpha}{P}\right)\ \cos\left(2\pi\frac{2\alpha}{P}\right)\ \ldots\ \cos\left(2\pi\frac{N\alpha}{P}\right)\right] \tag{4.13a}$$

$$\psi_s(\alpha) = \left[\sin\left(2\pi\frac{\alpha}{P}\right)\ \sin\left(2\pi\frac{2\alpha}{P}\right)\ \ldots\ \sin\left(2\pi\frac{N\alpha}{P}\right)\right] \tag{4.13b}$$

map the error signal and are based on the sine-cosine form of the Fourier series:

$$y(x) = a_0 + \sum_{n=1}^{N}\left(a_n\cos\left(2\pi\frac{n}{P}x\right) + b_n\sin\left(2\pi\frac{n}{P}x\right)\right) \tag{4.14}$$

Note that this is possible due to the periodic property of the error signal. Furthermore, the equal in length vector sets consist out of N sinusoids with a frequency interval of $\frac{1}{P}$. The corresponding parameter vectors θ_c and θ_s correspond to the weighting of the individual related sinusoids.

The amount of sinusoids and interval grid is determined by the quality of the resulting basis function fit. Figure 4.4 shows the effect of the number of sinusoids and interval choice for the error signal. Here, error signal (▪▪▪) that is obtained from the setup is used to map the occurring higher and lower periodic disturbances. Initially, there are 25 sinusoids (note, 50 in total since there are 2 sinusoidal types) each with an interval of 1. The resulting basis function fit is shown by Curve (▬). It is observed that this particular basis set combination is not able to map the error signal due to the initial curve that gives the fit an offset. Furthermore, it can be observed that the higher frequency behavior is mapped more accurately than the low frequencies and therefore the next step is to increase interval P by 1. In addition, the number of sinusoids is increased to 50 to keep the final sinusoid frequency at 25 [Hz]. As a result that the frequency range maps from $\frac{1}{2}$ [Hz] to 25 [Hz] in steps of a half Hertz. Curve (▬) shows the trajectory of the resulting basis functions fit. It can be observed that the effect of the interval increased the performance of the line fit. However, looking at the zoomed window of the trajectory, it can be observed that the basis function set is not able to fit the higher accurate enough. Therefore, the number of sinusoids is increased to 100 i.e. frequency range extension from 25 to 50 [Hz]. As a result that the final basis functions set (▬) is able to track the lower and higher frequency range behavior. When comparing the Root Mean Square (RMS) error of each fit, respectively being $\left\{5.95\times10^{-9},\ 1.78\times10^{-10},\ 2.75\times10^{-11}\right\}$ [m], it is concluded that the basis functions describe the error well and that the parameter selection of the final set gives a qualitative projection of the error.

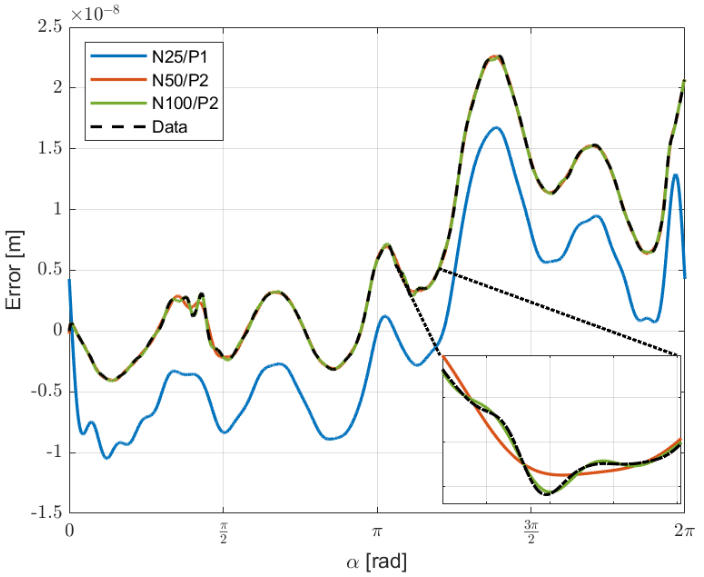


Figure 4.4: Basis function fit of the error signal (▪▪▪) with N number of sinusoid sets and P interval divider.

4.3 ILC with Hysteresis Compensation on Experimental Setup

The proposed ILC controller with basis functions is implemented on the system. The control architecture for the experiments is shown by Figure 4.2. During the experiments feedback controller C is taken as zero i.e. control output $u_j^c = 0$ for all trials j. Furthermore, two control cases are observed during experiments. During the first case, the ILC controller is implemented on the system without hysteresis compensation. In the second case, the ILC controller is implemented with hysteresis compensation obtained by the Generalized Prandtl-Ishlinskii compensator from Section 3.1.2. However, it is not possible to directly implement the compensator as implemented in Chapter 3 since the hysteresis compensator has a bounded reference as input and a bounded control output as output, whereas the reference in walking behavior is continuously increasing or decreasing. Therefore, the input of the compensator is scaled such that the input reference of a single shear step ($0 \leq y_{Si} \leq \frac{3.5}{2} \times 10^{-6}$ [m]) corresponds to the control input ($-250 \leq u_{Si} \leq 250$ [V]). Furthermore, the compensator input is connected to the output of the waveform scalar and the compensator output to the input of the plant.

The basis function set is defined by the final functions set as described by Section 4.2. Initial parameter vector θ_0 is defined so that the shear elements are actuated as normal shear waveforms (discussed in Section 1.3) without any interference of the error fit property of the basis function i.e. $\theta_c = \theta_s = 0$. As a result that the setup is actuated in a normal way since there is no feedback excitation by th feedback controller. In addition, the control input is not affected by disturbances caused by incorrect fitting of θ_c and θ_s during the initial trial.

The obtained feedforward signal is used for compensation by both shear elements. Therefore, the signal needs to be actuated by both shears and therefore need to be shifted in the regions where the clamps do not make contact with the mover such that (1.2) is still satisfied. Nest, shifted signals are added to the corresponding shear inputs which then result in enhanced shear inputs. Figure 4.5 shows the shifting of the basis functions fit (▪▪▪) and the resulting enhanced shear input waveforms (▬) and (▬). It can be observed that the individual compensation waveforms are shifted at regions (🟩) where the corresponding clamp element is not in contact with the mover. As a result, that both compensation waveforms start and end at zero voltage and therefore make sure that (1.2) still holds.

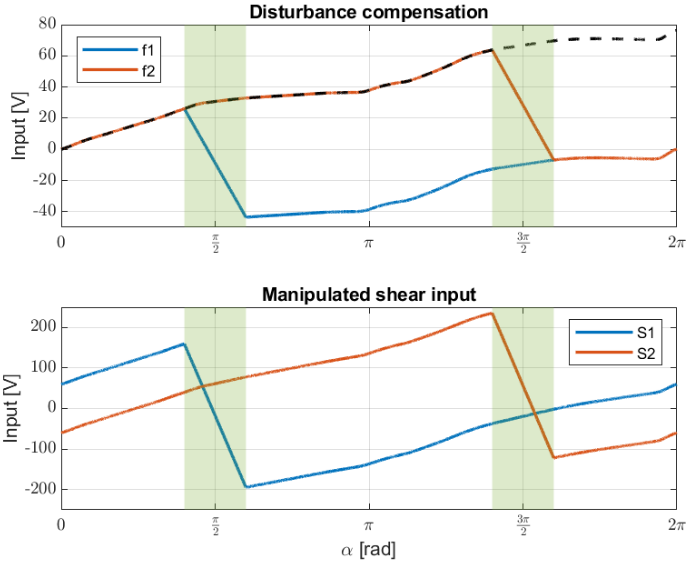


Figure 4.5: Shear waveform enhancement using basis function output where (🟩) are the regions where the clamp groups do not make contact with the mover. (top) shows the basis function fit (▪▪▪) and the shifted feedforward signal for shear set 1 (▬) and 2 (▬). (bottom) shows the resulting enhanced shear input.

Each iteration (or trial) consists out of the same procedure. First, the basis function set is constructed before the first trial. In addition, the initial parameter vector θ should be defined as discussed in the previous paragraph. Next, the first trial is executed and the data of the experiment is used to update the parameter vector with the basis function set. Finally, the basis function output is shifted so that the result can be applied to the shear input and therefore compensate for the error. The complete algorithm as discussed can be observed below.

Listing 4.1: Iteration procedure for ILC with basis fucntions

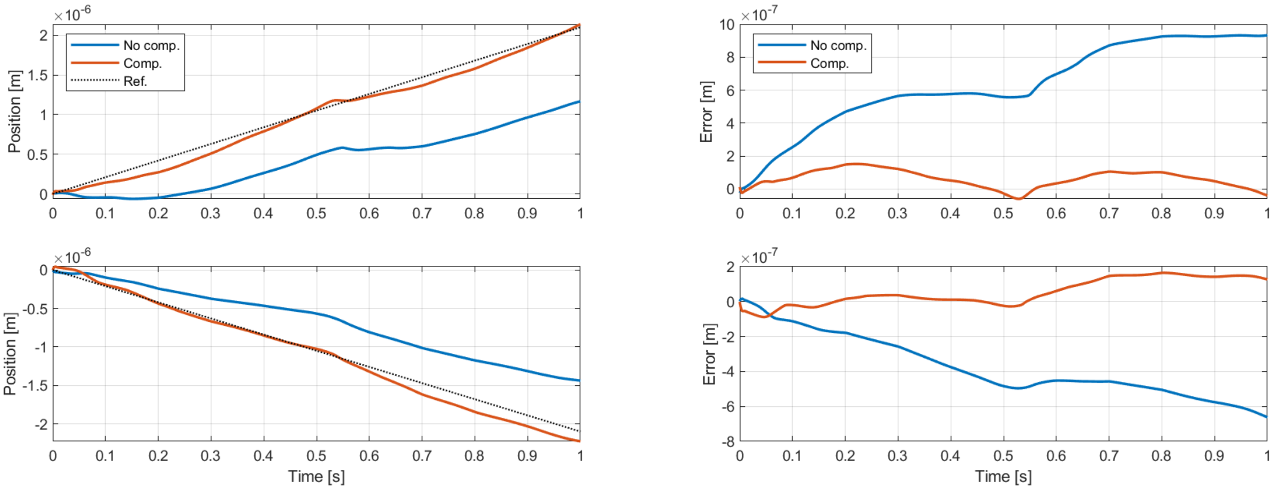
```
% Prior to first iteration
Define basis functions set $\Psi$
Set initial parameter vector $\theta$

% Iteration execution
for j=1:n
    Perform experiment for one step with control input $u_j$
    Find and update input parameters $\theta_{j+1}$ using least squares fit of the error
    Compute update law for next feedforward output $f_{j+1} = \psi\theta_{j+1}$
    Shift feedforward output for shear sets S1 and S2
    Update all control parameters and signals
end for
```

The ILC framework from algorithm 4.1 is implemented on the experimental setup and validated for a series of forward and backward walking experiments. The first experiment consists of the system without the ILC controller but with hysteresis compensation. The second experiment consists of the system with the ILC controller and without hysteresis compensation. The third and final experiment consists of the system with the ILC controller and hysteresis compensation. The RMS error values of each walking experiment with a drive frequency of 1 [Hz] are shown in Table 4.1.

Table 4.1: RMS error of the system in 1 [Hz] walking situation with and without ILC controller and hysteresis compensation.

Hysteresis compensation	ILC	Direction	RMSE [nm]
No	No	Positive	4780.372
Yes	No	Positive	67.420
No	No	Negative	2810.870
Yes	No	Negative	1618.861
No	Yes	Positive	14.126
Yes	Yes	Positive	5.751
No	Yes	Negative	5.589
Yes	Yes	Negative	5.445

The results of the walking experiments with and without hysteresis compensation and without ILC are shown in Figure 4.6a and Figure 4.6b. Here, the top and bottom figures respectively show the positive and negative direction movement. It can be observed that the hysteresis compensator decreases the tracking error while in walking behavior compared to the uncontrolled situation. This is also visible by the RMS error which decreases by almost a factor of 1.141×10^{-2}. Furthermore, it can be seen that the resulting error still has hysteresis behavior in it due to modeling errors in the compensator model. Moreover, the RMS error of the compensated situation in the negative direction is bigger compared to the forward direction since the compensator model is based on the positive direction data. This is an effect of the difference in clamping voltage for the actuation direction which results in a longer stroke as discussed in Chapter 2. Nevertheless, the RMS values show that the addition of the hysteresis compensator increase system performance.

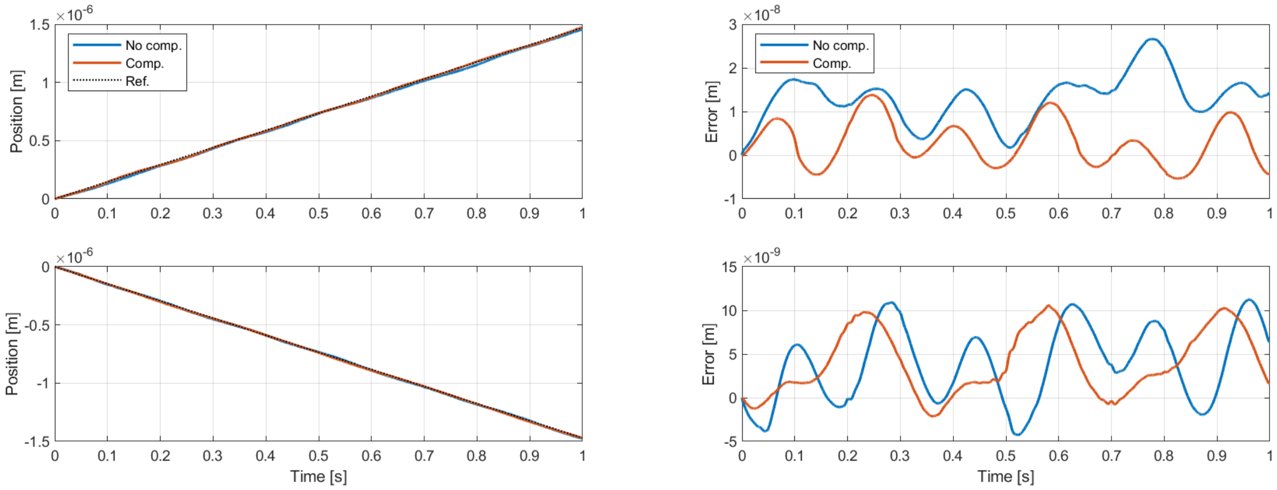


(a) Position

(b) Error

Figure 4.6: System output while in 1 [Hz] positive (top) and negative (bottom) direction walking situation with (▬) and without (▬) hysteresis compensation and without ILC where (a) the position of the mover and (b) the error with (┈) the reference input.

The results of the walking experiments with and without hysteresis compensation and with ILC are shown in Figure 4.7a and Figure 4.7b. Here, the top and bottom figures respectively show the positive and negative direction movement. For comparison, the ILC system output after 5 trials with a reference input of 1 [Hz] is used. It can be observed that the addition of the ILC controller significantly improves the performance of the system with and without the hysteresis compensation compared to the system response without ILC. In addition, the performance improvement is also visible for negative directions. Likewise with the system response without ILC, the system response including the hysteresis compensator result in a lower RMS error compared to the situations without compensation. The resulting residual error, shown in Figure 4.7b, consists of a multi-sine that is not periodic within the period of the actuation waveform. Therefore, another performance increase could be obtained by implementing projection-based basis functions as proposed in [58]. All things considered, the system response for the situations with ILC and hysteresis compensation for both directions result in an approximate RMS error decrease factor of 1.203×10^{-3} for the positive direction and 1.139×10^{-3} for the negative direction compared to the uncontrolled and uncompensated situation. However, the experiment results are based on a fixed drive frequency on which the compensator is modeled and does not validate for other frequencies outside the compensator scope.

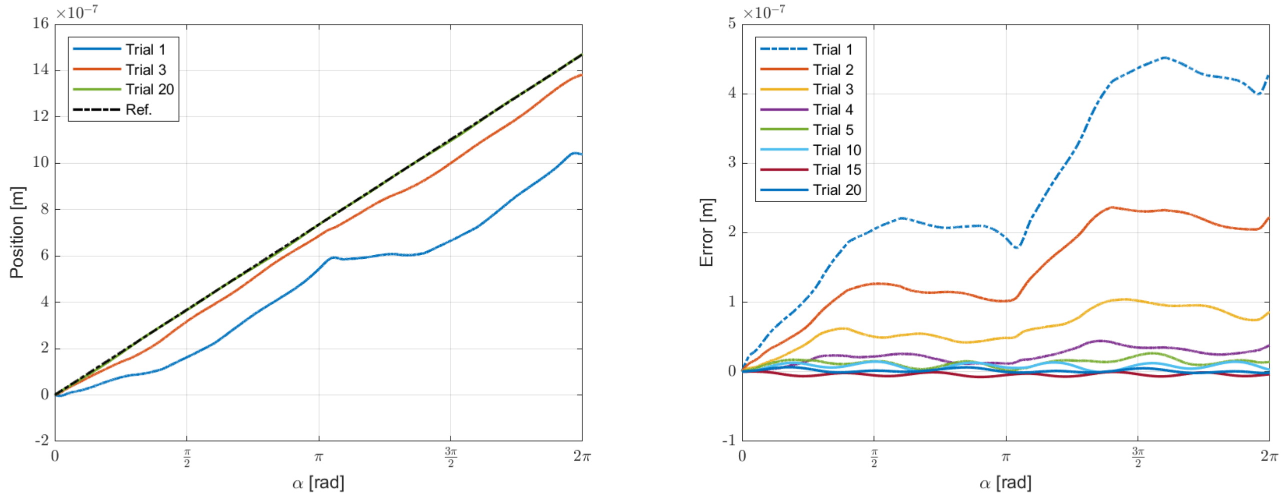


(a) Position

(b) Error

Figure 4.7: System output while in 1 [Hz] positive (top) and negative (bottom) direction walking situation with (━) and without (━) hysteresis compensation and with ILC where (a) the position of the mover and (b) the error with (┈) the reference input.

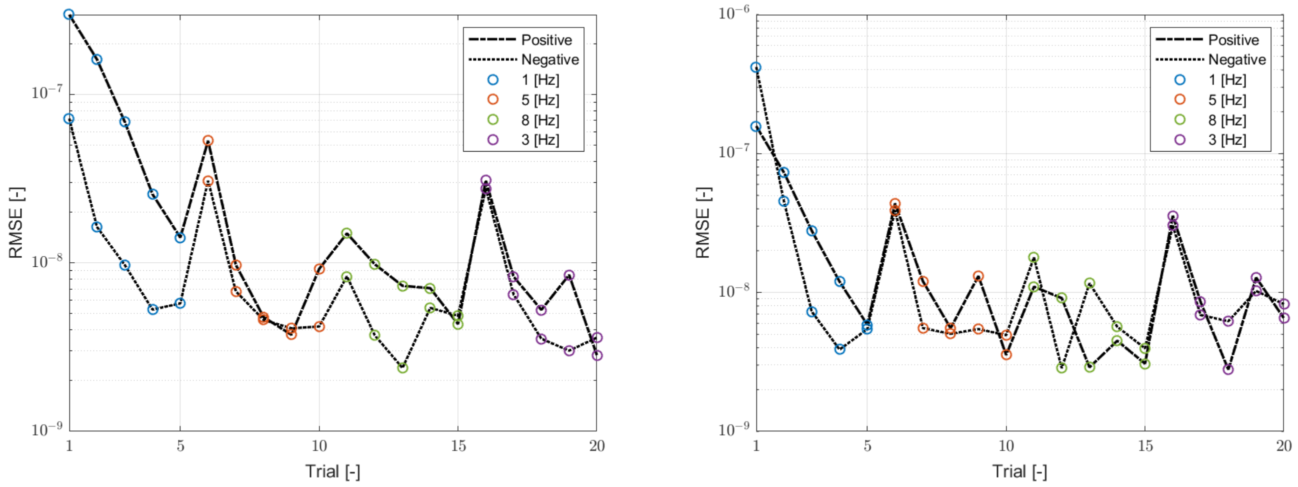
The main goal of introducing basis functions is that the system is able to reduce transient errors when changing the input reference for systems with repetitive tasks. To validate this property, four experiments are executed consisting of ILC control with and without hysteresis compensation in both walking directions. Each ILC experiment iterates through a set of 20 trials with varying drive frequencies in series of 5 trials with frequency order $f_d = \{1,\ 5,\ 8,\ 3\}$ [Hz]. The position and error evolution during the individual trials for the uncompensated hysteresis situation is respectively shown in Figure 4.8a and Figure 4.8b. In the error figure, the converging reduction property of the ILC controller can be observed. Furthermore, it can be observed that the position closely follows the reference input after five trials since the trend in the error curve is decreased to approximately zero. The RMS error of each trial is investigated to measure the overall controller performance for all four experiment situations. The resulting RMS error curve for the uncompensated situation is shown in Figure 4.9a and for the compensated situation is in Figure 4.9b. The RMS error of each trial for the uncompensated situation in both directions shows that the error decreases over each trial. Furthermore, it can be seen that the system needs a single trial to decrease the transient error that occurs when changing the drive frequency. Moreover, the RMS error curve for the compensated situation is more fluctuating outside the 1 [Hz] region compared to the uncompensated situation. This is a direct effect of the hysteresis compensator not being able to handle rate-dependent behavior that occurs in the system. To fully test and guarantee the stability of the compensated situation, the extension to a rate-dependent compensator model should be considered.



(a) Position

(b) Error

Figure 4.8: ILC iteration output without hysteresis compensation where (a) the position of the mover and (b) the error with (⋯) the reference input.



(a) wo. compensation

(b) w. compensation

Figure 4.9: Root-Mean-Square error evolution of system with ILC controller for 20 trials (a) without hysteresis compensation and (b) with hysteresis compensation where (▪▪▪) is the movement in positive direction and (⋯) in negative direction.

Chapter 5

Conclusion & Future Work

In this chapter, the conclusions and recommendations of this research is discussed. First, conclusions are discussed about the obtained results and observations. Secondly, recommendations are discussed that propose possible future steps within the research problem.

5.1 Conclusions

This research focused on the control of a hysteric piezo actuator. Here, stacked sets of polarized piezo elements are actuated by predefined waveforms such that the resulting translation outputs a walking movement on the mover element surface and therefore moves the mover element. Open-loop experiments showed that multiple performances decreasing phenomena such as creep and hysteresis are acting on the system. Furthermore, so-called liftoff and contact experiments showed that the actuating direction of the piezo elements results in different output behavior.

The main goal of this research was to compensate for the hysteresis effect with the use of a feed-forward compensator based on the inverse Prandtl-Ishlinskii Model. As an extension, variations on the classical Prandtl-Ishlinskii model were proposed. This included the introduction of a generalized envelope function for the play operator, a memory element, and a neural network adaption of the PI model. The compensators are implemented and tested on the shear elements for performance analysis. Experimental data showed that the addition of the compensators results in a significant performance increase compared to the uncompensated situation. However, tracking performance decreased at the regions where the input rate changes direction (turn-back points). This is a result of model errors that are introduced in the process of obtaining the model parameters and inverse. Moreover, comparing all the compensators showed that the general performance is similar. Furthermore, it showed that the generalized Prandtl-Ishlinskii model (GPI) performed the best. However, the memory element model (MEMS) was implemented with half the number of elements as the other models. Therefore, when equalizing the number of elements, the MEMS model would outperform the other models in their current construction. In contrast with all the compensator models, the neural network model (PI-NN) performs the worst. This is a result of the lack of a direct inverse and the method used to overcome this issue. In addition, the cross structure incompatibility of the Pytroch model structure and the Simulink control structure.

The GPI compensator is implemented on each shear element and tested in a walking situation. In addition, an ILC controller with basis functions is introduced to increase performance. The introduction of basis functions reduce the transient error that occurs when changing the reference input during trials of the repetitive system. The constructed basis function set consist out of two sets. The first set is used to scale the shear waveforms. The second set is based on the Fourier series of the measured disturbances. Furthermore, the resulting basis function waveform output is shifted during the moments the clamp does not make contact with the mover such that the input rate of the shear elements are equal. The ILC controller is implemented and tested with and without hysteresis compensation. Experimental results for the 1 [Hz] drive frequency case showed that the performance significantly increased compared to the uncompensated and uncontrolled. In addition, the performance with ILC and compensation decreased the performance difference between the positive and negative direction. When comparing the results for varying drive frequencies, it can be observed from the RMS error curve that the basis functions reduce the transient error during frequency changes and converge to the reference input. Furthermore, the RMS error curve for the compensated situation is more fluctuating outside the 1 [Hz] region compared to the uncompensated situation. This is a direct effect of the hysteresis compensator not being able to handle rate-dependent behavior that occurs in the system. However, a complete analysis of the overall stability for the system with ILC and hysteresis compensation with a varying reference input could not be executed since the incapability of the hysteresis compensator to handle rate-dependent behavior.

All things considered, the system response with a 1 [Hz] drive frequency for the situations with ILC and hysteresis compensation for both directions result in an approximate RMS error decrease factor of 1.203×10^{-3} for the positive direction and 1.139×10^{-3} for the negative direction compared to the uncontrolled and uncompensated situation.

5.2 Future work

Throughout this report, several developments and issues can be recommended for future research on the research problem:

- *Improvement parameter fitting algorithm for PI models.*
  As the results from the PI and GPI models in Section 3.2 showed, the obtained model still lacks accuracy. For this research, the parameters are obtained through a least square problem. This results in a decrease in accuracy since the parameter fit lacks precision. In literature, a variety of parameter fitting algorithms for hysteresis models are proposed. Results showed that a higher model accuracy could be achieved by implementing a different more advanced optimization solver. Therefore, to increase model accuracy, parameter fitting should be improved by introducing a different more advanced solver such as a modified particle swarm optimization algorithm [59].

- *Extension rate-dependent mapping for PI models.*
  The implemented PI and GPI models are based upon the standard framework which explained in 3.1.1 is rate-independent. Therefore it is not possible to map rate-dependent behavior that occurs. This is also what is observed when compensating in the walking situation with changing drive frequencies as discussed in Section 4.3. Since the results in Chapter 3 showed that the hysteresis behavior is rate-dependent, performance increase can be achieved by extending the PI and GPI model with the compatibility to handle rate-dependent hysteresis behavior.

- *Further research on inverse PI Neural Network.*
  As discussed in Section 3.1.4, the implemented inverse model is not the real inverse but an approximation based on experimental data of an ILC controller. This resulted in a decrease

in performance. Moreover, the Simulink control structure was not compatible with the constructed PuTorch neural network structure since Matlab does not fully support the ONNX exchange language. Therefore, the control signal used in the experiments was in the form of a fixed output sequence. This limited the flexibility during experiments since the reference could not be changed. The forward model showed promising results based on its generic design and approach with pre-determined Prandtl operators. Since it was not possible to investigate the full potential of the compensating model i.e. rate dependencies, further research could be executed.

- *Implementation projection ILC w. basis functions in time-domain.*
  The ILC implementation of the basis functions is achieved by mapping the resulting basis function fit of the error to the α-domain. During implementation, the goal was to implement the basis functions as a projection running in the time-domain [58]. Early implementations showed that performance increase was achieved due to the reduction of the periodic disturbances and resulting non-periodic disturbances. However, due to hardware limiting factors of the experimental setup, this could not be implemented for multiple trials with the current set of basis functions. In addition, (1.2) needs to hold, therefore the control signals need to be shifted when the clamps are not in contact with the mover. This introduced another issue due to the continuous projection of the disturbances. Since early implementation showed promising results with both performance and generic implementation, the implementation of projection-based ILC with basis function should be further research.

Appendix A

Encoder Calibration

The used encoder is a sin/cos encoder that uses two sinusoidal differential signals for measurement values. The encoder structure consists out of a readhead and a grating strip that makes use of the Moiré effect [60]. The read head emits light to the grating strip which reflects parts of it into the reading sensor of the readhead. By doing so, a sine and cosine signal with voltage offsets is generated. The sinusoidal signals are outputted as a pair of original and negated signals to create differential signaling. Moreover, differential signaling is used to remove disturbances on the signal that are created by the environment, such as electromagnetic compatibility. The signal pairs are processed by the real-time target to retrieve the current position of the mover. Here the first step is to do a summation over the signal pairs. This translates the two signal pairs back into the normal sine and cosine signals without an offset and additional disturbances due to signal transfer. The position is determined by counting the periods from the sinusoidal signals. By only doing this, a staircase curved position output is obtained. To obtain a better approximation of the real position between period counts, an interpolation correction is added. This is done by determining the inverse tangent of the two sinusoidal signals and adding this value to the period count.

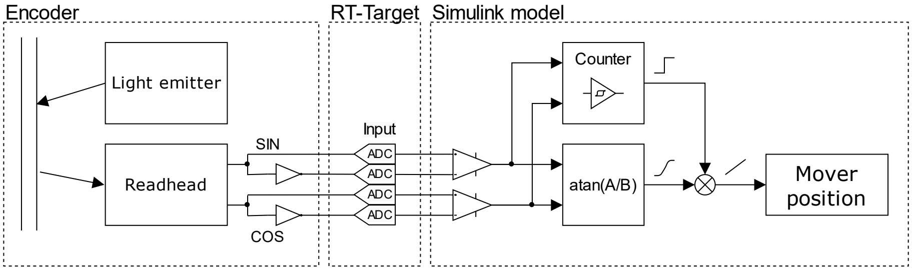


Figure A.1: Block scheme of mover position process from encoder data.

The encoder signals have an additional error due to misalignment and mechanical tolerances of the construction. When a Lissajous plot is made with the sine and cosine data, it can be seen in Figure A.2 that the resulting curve (━) is not a circle but an decreased oval, i.e. the signals do not have the same amplitude and correct phase. To overcome this error, and therefore increase the accuracy, an additional calibration is carried out. This calibration is constructed in the form of a software correction that will handle the raw encoder signals. The calibration method is based on a Lissajous correction method [61]. Here, use is made of the knowledge of how the two signals should be when calibrated correctly. When combining these two theoretical equations and convert them to a least square problem, (A.1) can be obtained. Here, u_1 and u_2 represents the calibrated sinusoidal encoder signals.

$$(u_1 - p)^2 + \left[\frac{(u_2 - q)r + (u_1 - p)\sin(\alpha)}{\cos(\alpha)}\right]^2 = R^2 \tag{A.1}$$

in the form,

$$Au_1^2 + Bu_2^2 + Cu_1u_2 + Du_1 + Eu_2 = 1 \tag{A.2}$$

where,

$$\begin{aligned} A &= \left(R^2\cos^2(\alpha) - p^2 - r^2q^2 - 2rpq\sin(\alpha)\right)^{-1} \\ B &= Ar^2 \\ C &= 2Ar\sin(\alpha) \\ D &= -2A(p + rq\sin(\alpha)) \\ E &= -2Ar(rq + p\sin(\alpha)) \end{aligned} \tag{A.3}$$

The found coefficients are used for determining the correction parameters as described by (A.4). The corrected encoder signals u_1^* and u_2^* are determined by (A.5). Figure A.2 shows the calibrated encoder signal (▬) of the experimental setup with its calibration goal (┈).

$$\begin{aligned} \alpha &= \arcsin\left(\frac{C}{\sqrt{4AB}}\right) \\ r &= \sqrt{\frac{B}{A}} \\ p &= \frac{2BD - EC}{C^2 - 4AB} \\ q &= \frac{2AE - DC}{C^2 - 4AB} \end{aligned} \tag{A.4}$$

$$\begin{aligned} u_1^* &= u_1 - p \\ u_2^* &= \frac{1}{\cos(\alpha)}\left[(u_1 - p)\sin(\alpha) + r\,(u_2 - q)\right] \end{aligned} \tag{A.5}$$

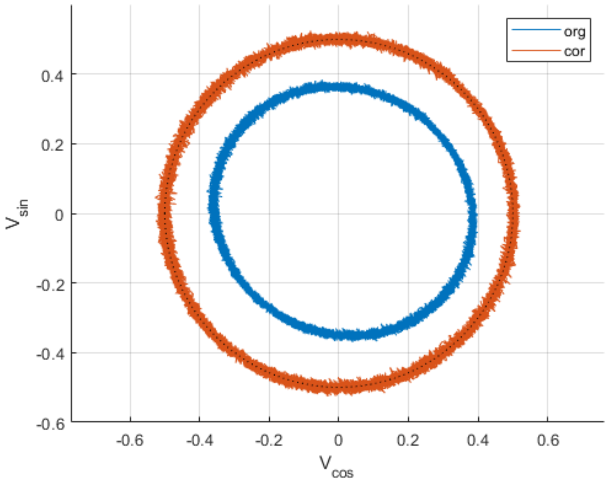


Figure A.2: Encoder calibration process via Lissajous correction, where (🟦) is the uncalibrated encoder data, (🟧) is the calibrated encoder data, and (⋯) is the calibration goal.

Appendix B

Open-Loop Experiments

B.1 Clamp Liftoff & Contact Moments

The clamp liftoff and contact moments play an important role in the walking behavior of the piezo actuator. Liftoff is defined as the first moment in a drive cycle where the clamp stops having contact with the mover. Opposite to liftoff, contact is defined as the first moment in a drive cycle where the clamp comes in contact with the mover. A liftoff/contact experiment is executed in order to investigate the moments these actions occur where the exact moment is portrayed by the current clamp voltages. In this experiment, a clamp set is actuated with a low frequency sawtooth waveform while the corresponding shear set is actuated with a faster sinusoidal waveform. The other clamp and shear set are stationary while the experiment is executed.

The velocity output of the mover is investigated to get a better estimation of the liftoff and contact points since the position is affected by drift. The contact moments are determined when the velocity of the mover starts to oscillate. Liftoff moments are determined when the velocity of the mover is stopped oscillating. Figure B.1 shows the filtered velocity output of the setup when the clamps are excited with a waveform frequency of 1 [Hz] (top figure) and -1 [Hz] (bottom figure). The scaled-down clamp waveform is depicted by (▪▪▪) and the liftoff and contact moments are respectively depicted by (△) and (▽). The first observation is that the contact and liftoff moments are not at an equal voltage. This could be an effect of imperfections in the internal mechanics and hysteresis in the clamp elements. Furthermore, the experimental data indicate that the contact and liftoff moments occur in a repeatable manner within the same actuation frequency. However, contact and liftoff voltages differ when comparing the average value over a set of actuation frequencies. In addition, differences are observed between the positive and negative actuation frequencies.

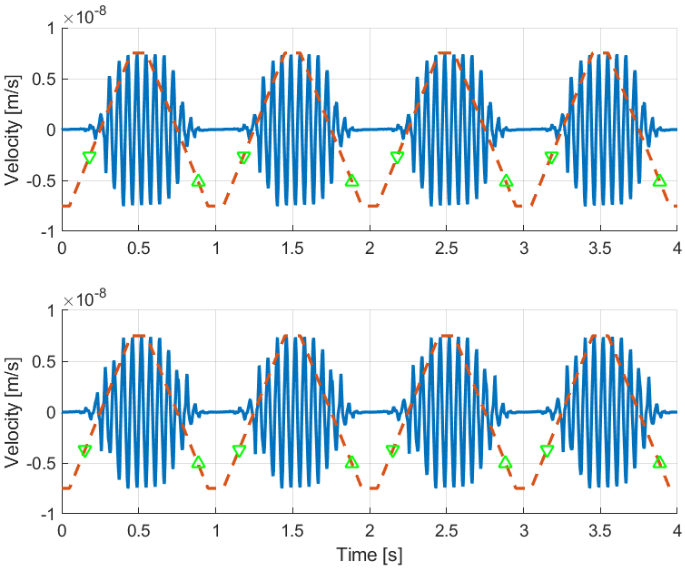
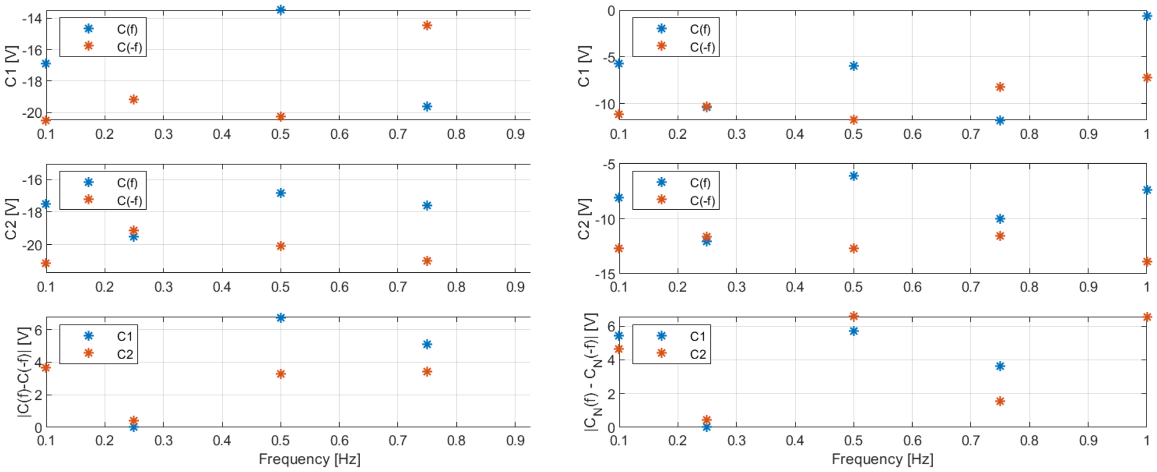


Figure B.1: Velocity output of the liftoff/contact experiment with (top) 1 [Hz] and (bottom) -1 [Hz] clamp actuation direction. (▪ ▪ ▪) depicts the scaled down clamp waveform, and the liftoff and contact moments are respectively depicted by (△) and (▽).

The average contact and liftoff voltages for a set of clamp actuation frequencies are determined and compared. Figure B.2 shows the (a) clamp experiment and the (b) liftoff experiment with clamp actuation frequency $f_c \in \{0.1, 0.25, 0.5, 0.75, 1\}$ [Hz]. The top and middle figures show the clamp voltage for respectively clamp set 1 and set 2 with ($*$) positive and ($*$) negative frequency. The first observation is that the figures indicate a difference between contact and liftoff voltages for different actuation frequencies. Furthermore, it is observed that there is a varying difference between the positive and negative frequency voltages. In most frequency cases the data shows that the contact and liftoff moments happen at a lower voltage. As a result that the clamp elements are longer in contact with the mover and therefore produce a longer stroke. The bottom figure shows the absolute voltage difference between the positive and negative frequency for clamp set 1 ($*$) and 2 ($*$). To achieve a reproducible motion for positive and negative actuation frequency, the voltage difference should be zero. However, this is not the case for the experimental setup and hence the difference in walking behavior as observed in Figure 2.4a.



(a) Contact

(b) Liftoff

Figure B.2: (a) clamp experiment and (b) liftoff experiment with clamp actuation frequency $f_c \in \{0.1, 0.25, 0.5, 0.75, 1\}$ [Hz] where, (top) and (middle) are the clamp voltage for respectively clamp set 1 and set 2 with ($*$) positive and ($*$) negative frequency, and (bottom) the absolute difference in voltage between the positive and negative frequency for clamp set 1 ($*$) and 2 ($*$).

B.2 Creep

Creep is the phenomenon where the actuator drifts slowly after an increase of the input for a constant applied electric field [8]. Creep reacts after the dynamic response of the system and settles in a logarithmic shape over time. Creep is defined by (B.1) where $L(t)$ is the resulting displacement, L_o is a nominal constant displacement value after 0.1 seconds, γ is the creep factor that determines the logarithmic growth rate. The growth factor γ is depending on the amount of input voltage applied to the piezo element. Figure B.3 shows the resulting creep phenomena on shear set 1 for a step input with amplitudes $A \in \{25, 50, 100, 150\}$ [V]. Here, it can be observed that creep is indeed behaving in a logarithmic way and that the amount of creep is depending on the amount of input difference on the piezo element. The creep phenomenon is out of scope and therefore will not be modeled.

$$L(t) = L_o \left[1 + \gamma \log_{10} \left(\frac{t}{0.1} \right) \right] \tag{B.1}$$

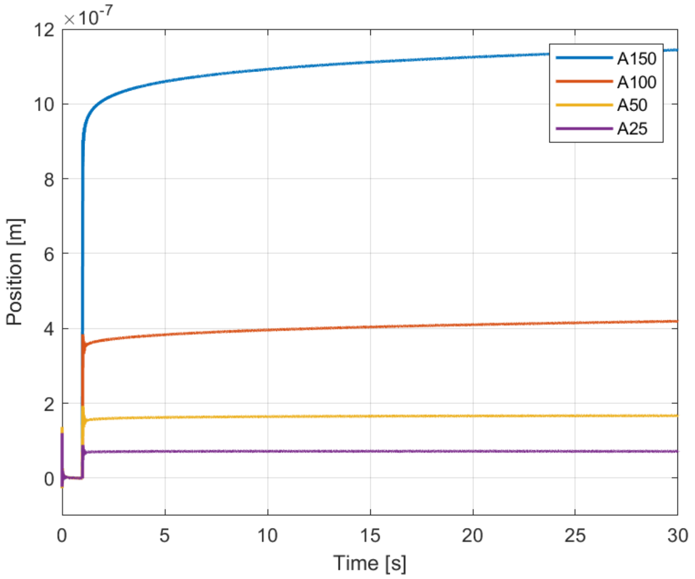
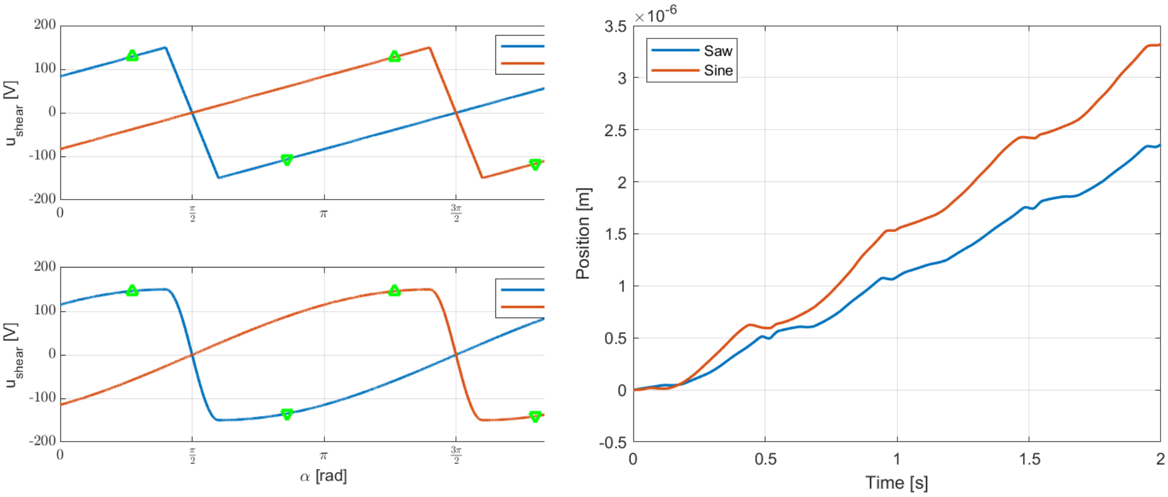


Figure B.3: Creep phenomena after a step input on shear set 1 for input amplitudes $A \in \{25, 50, 100, 150\}$ [V].

B.3 Input Waveform Type

The input waveform plays an important role in the motion performance of a piezo actuator. A piezo actuator can be driven with different waveforms based on the desired performance e.g. velocity, motion, and force. Three common basic waveform types [9] are: square wave, sawtooth, and sinusoidal. A sawtooth waveform, as shown in 1.4a for the shear elements, was used in previous research on the experimental setup [10][11]. In addition to the fixed type of waveforms, methods such as waveforms generated by model/data-based optimization [12], iterative learning-based [13], or force-based [14] are also possible. For the scope of this research, only the static sawtooth and sinusoidal waveforms are discussed in this section.

Sawtooth and sinusoidal waveforms are constructed for the experimental setup to compare the effects on the performance in walking situation. In [9] a square, sawtooth, and sinusoidal waveform are generated for an inchworm actuator on a miniature robot. This study showed that the sinusoidal and triangular waveforms performed better than the square waveform. This was due to the incapability of the square wave to change the direction of the robot. Furthermore, the sinusoidal waveform gave an increase in translational displacement compared to the sawtooth. This result forecasts an expected performance increase while walking when applying a sinusoidal waveform instead of a sawtooth waveform on the shear elements. The waveforms for this research are constructed with a period from 0 to 2π [rad] to be dependent on the commutation angle and driven by the drive frequency. For the experiment, the sawtooth waveform from 1.3 is adopted. Furthermore, the sinusoidal waveform is constructed of two parts of unequal frequency sinusoids such that it is possible to reset the shear element when the corresponding clamp does not make contact with the mover. Figure B.4a shows the constructed sawtooth and sinusoidal input waveform. Furthermore, Figure B.4b shows the resulting mover position when actuating the shear elements with the constructed waveforms in the walking situation with a drive frequency of 1 [Hz]. It can be observed that the position curve of the sinusoidal waveform resulting in a longer more bouncier stroke compared to the sawtooth waveform. This is an effect of the sinusoidal waveform slightly compensating for some hysteresis behavior that occurs in the piezo element. The sinusoidal waveform when in contact with the mover has similar curves as the resulting hysteresis compensator output which is revealed in the next chapter. In addition, the liftoff and contact moments occur at a different shear voltage compared to the sawtooth waveform. This results into a longer contact time with the mover and therefore into a longer stroke.



(a) Shear input

(b) Position

Figure B.4: (a) Constructed (top) sawtooth and (bottom) sinudsoidal shear input waveform in commutation angle domain where (△) and (▽) are respectively the liftoff and contact moments. (b) Mover position output while in walking situation with sawtooth and sinudsoidal shear input waveforms and a drive frequency of 1 [Hz].

Bibliography

[1] X. Yang, Y. Guan, and Z. Zhang. “Robust tracking control of an XY compliant nanomanipulator with variable loads”. In: *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. 2018, pp. 1–6.

[2] Z. Hu, G.P. Maul, and D. Farson. “Piezo actuated vibratory feeding with vibration control”. In: *International Journal of Production Research* 45.5 (2007), pp. 1089–1100.

[3] S. Mazerolle et al. “Positioning, Handling and Measuring inside a Scanning Electron Microscope”. In: *Journées du RTP Microrobotique* (2003).

[4] Y. Kurita et al. “Piezoelectric Tweezer-Type End Effector With Force- and Displacement-Sensing Capability”. In: *IEEE/ASME Transactions on Mechatronics* 17.6 (2012), pp. 1039–1048.

[5] PI Ceramic GmbH. *Displacement Modes of Piezoelectric Actuators*. 2021. URL: `https://www.piceramic.com/en/piezo-technology/properties-piezo-actuators/displacement-modes/`.

[6] P. J. M. M. Tacx. “Performance Analysis and Feasibility of Advanced Control for a Walking Piezo Actuator”. Internship project report. Eindhoven University of Technology, Dec. 3, 2018.

[7] R. J. E. Merry et al. “Using a Walking Piezo Actuator to Drive and Control a High-Precision Stage”. In: *IEEE/ASME Transactions on Mechatronics* 14.1 (2009), pp. 21–31.

[8] H. Jung and D. Gweon. “Creep characteristics of piezoelectric actuators”. In: *Review of Scientific Instruments* 71 (Apr. 2000), pp. 1896–1900.

[9] T. Kusakawa et al. “Control waveforms applied to piezo elements used in a miniature robot”. In: *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004*. 2004, pp. 307–312.

[10] L. Aarnoudse et al. “Commutation-Angle Iterative Learning Control for Intermittent Data: Enhancing Piezo-Stepper Actuator Waveforms”. In: *ArXiv* abs/2006.13572 (2020).

[11] N. Strijbosch and T. Oomen. “Hybrid-MEM-Element Feedforward: With Application to Hysteretic Piezoelectric Actuators”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. 2020, pp. 934–939.

[12] R. J. E. Merry et al. “Modeling and Waveform Optimization of a Nano-motion Piezo Stage”. In: *IEEE/ASME Transactions on Mechatronics* 16.4 (2011), pp. 615–626.

[13] N. Strijbosch et al. “Commutation Angle Iterative Learning Control: Enhancing Piezo-Stepper Actuator Waveforms”. In: *IFAC-PapersOnLine* 52.15 (2019), pp. 579–584.

[14] F. Szufnarowski and A. Schneider. “Two-dimensional dynamics of a quasi-static legged piezoelectric actuator”. In: *Smart Materials and Structures - SMART MATER STRUCT* 21 (May 2012).

[15] Dong An et al. “Compensation of Hysteresis on Piezoelectric Actuators Based on Tripartite PI Model”. In: *Micromachines* 9 (2018).

[16] Dragan Damjanovic. “Ferroelectric, dielectric and piezoelectric properties of ferroelectric thin films and ceramics”. In: *Reports on Progress in Physics* 61.9 (Sept. 1998), pp. 1267–1324.

[17] M. Brokate and J. Sprekels. *Hysteresis and Phase Transitions.* Vol. 121. Applied Mathematical Sciences. Springer, 1996.

[18] Ping Ge and Musa Jouaneh. “Modeling hysteresis in piezoceramic actuators”. In: *Precision Engineering* 17.3 (1995), pp. 211–221.

[19] F. Weibel et al. “Real-time compensation of hysteresis in a piezoelectric-stack actuator tracking a stochastic reference”. In: *2008 American Control Conference.* 2008, pp. 2939–2944.

[20] M. Fuad Mohammad Naser and F. Ikhouane. “Characterization of the Hysteresis Duhem Model”. In: *IFAC Proceedings Volumes* 46.12 (2013), pp. 29–34.

[21] M. Goldfarb and N. Celanovic. “Modeling piezoelectric stack actuators for control of micromanipulation”. In: *IEEE Control Systems Magazine* 17.3 (1997), pp. 69–79.

[22] Y. Liu et al. “Modelling and compensation of hysteresis in piezoelectric actuators based on Maxwell approach”. In: *Electronics Letters* 52.3 (2016), pp. 188–190.

[23] M. Ventra, Y. Pershin, and L. Chua. “Circuit Elements With Memory: Memristors, Memcapacitors, and Meminductors”. In: *Proceedings of the IEEE* 97 (2009), pp. 1717–1724.

[24] W. Ramberg and W. R. Osgood. “Description of Stress-Strain Curves by Three Parameters”. In: 1943.

[25] J. Setz. “Modeling hysteresis in a walking piezo motor”. Internship project report. Eindhoven University of Technology, Nov. 19, 2019.

[26] P. Krejci and K. Kuhnen. “Inverse control of systems with hysteresis and creep”. In: *IEE Proceedings - Control Theory and Applications* 148 (3 May 2001), 185–192(7).

[27] M. F. Al Janaideh. “Generalized Prandtl-Ishlinskii hysteresis model and its analytical inverse for compensation of hysteresis in smart actuators”. 2009. URL: `https://spectrum.library.concordia.ca/976620/`.

[28] J. Zhang et al. “Optimal Compression of a Generalized Prandtl-Ishlinskii Operator in Hysteresis Modeling”. In: vol. 3. Oct. 2013.

[29] M. Al Janaideh et al. “Inverse generalized asymmetric Prandtl-Ishlinskii model for compensation of hysteresis nonlinearities in smart actuators”. In: *2009 International Conference on Networking, Sensing and Control.* 2009, pp. 834–839.

[30] T. Oomen. “Control for precision mechatronics”. In: *Encyclopedia of Systems and Control.* Springer, 2019. ISBN: 978-1-4471-5057-2.

[31] N. Strijbosch, K. Tiels, and T. Oomen. “Hysteresis Feedforward Compensation: A Direct Tuning Approach Using Hybrid-MEM-Elements”. In: *IEEE Control Systems Letters* 6 (2022), pp. 1070–1075.

[32] N. Strijbosch and T. Oomen. “Hybrid-MEM-Element Feedforward: With Application to Hysteretic Piezoelectric Actuators”. In: *2020 59th IEEE Conference on Decision and Control (CDC).* 2020, pp. 934–939.

[33] N. Sharma, V. Jain, and A. Mishra. “An Analysis Of Convolutional Neural Networks For Image Classification”. In: *Procedia Computer Science* 132 (2018), pp. 377–384.

[34] A. Narayan and R. Muthalagu. “Image Character Recognition using Convolutional Neural Networks”. In: *2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)*. 2021, pp. 1–5.

[35] D. Nagajyothi and P. Siddaiah. “Speech Recognition Using Convolutional Neural Networks”. In: *International Journal of Engineering & Technology* 7.4.6 (2018).

[36] Y. Liu et al. “Adaptive Neural Network Control for Active Suspension Systems With Time-Varying Vertical Displacement and Speed Constraints”. In: *IEEE Transactions on Industrial Electronics* 66.12 (2019), pp. 9458–9466.

[37] Y. Song, Y. Zhou, and R. Han. *Neural networks for stock price prediction.* 2018. arXiv: `1805.11317 [q-fin.ST]`.

[38] G. Li et al. “Neural-network-based modelling and analysis for time series prediction of ship motion”. In: *Ship Technology Research* 64.1 (2017), pp. 30–39.

[39] R. Pascanu, T. Mikolov, and Y. Bengio. “On the difficulty of training recurrent neural networks”. In: vol. 28. 3. 2013, pp. 1310–1318.

[40] C. Nwankpa et al. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning.* 2018.

[41] J. Sola and J. Sevilla. “Importance of input data normalization for the application of neural networks to complex industrial problems”. In: *IEEE Transactions on Nuclear Science* 44.3 (1997), pp. 1464–1468.

[42] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. In: *Nature News* 323 (1986), pp. 533–536.

[43] B. Polyak. *Introduction to Optimization.* Optimization Software, Inc., July 1987.

[44] J. Hu, Y. Zhong, and M. Yang. “Hysteresis modeling of piezoelectric micro-positioning stage based on convolutional neural network”. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 235.2 (2021), pp. 170–179.

[45] C. Grech et al. “Dynamic Ferromagnetic Hysteresis Modelling Using a Preisach-Recurrent Neural Network Model”. In: *Materials* 13.11 (2020).

[46] PyTorch. *PyTorch: From research to production.* 2021. URL: `https://pytorch.org/`.

[47] M.D. Zeiler et al. “On rectified linear units for speech processing”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing.* 2013, pp. 3517–3521.

[48] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, Ltd, 2000.

[49] X. Glorot and Y. Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.* Vol. 9. Proceedings of Machine Learning Research. PMLR, 2010, pp. 249–256.

[50] D.A. Bristow, M. Tharayil, and A.G. Alleyne. “A survey of iterative learning control”. In: *IEEE Control Systems Magazine* 26.3 (2006), pp. 96–114.

[51] M. Q. Phan et al. “Robustification of iterative learning control and repetitive control by averaging”. In: *International Journal of Control* 86.5 (2013), pp. 855–868.

[52] x. Li et al. “Welding Process Tracking Control Based on Multiple Model Iterative Learning Control”. In: *Mathematical Problems in Engineering* 2019 (May 2019).

[53] L. Blanken and T. Oomen. “Multivariable Iterative Learning Control Design Procedures: From Decentralized to Centralized, Illustrated on an Industrial Printer”. In: *IEEE Transactions on Control Systems Technology* 28.4 (2020), pp. 1534–1541.

[54] F. Boeren et al. "Unified ILC framework for repeating and varying tasks: A frequency domain approach with application to a wire-bonder". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015, pp. 6724–6729.

[55] T. Oomen. "Learning for Advanced Motion Control". In: *International Workshop on Advanced Motion Control*. 2020, pp. 65–72.

[56] J. C. D. van Zundert et al. "Resource efficient ILC: enabling large tasks on an industrial position-dependent flatbed printer". In: *IFAC-PapersOnLine* 49 (2016), pp. 567–574.

[57] M.Q. Phan and J.A. Frueh. "Learning control for trajectory tracking using basis functions". In: *Proceedings of 35th IEEE Conference on Decision and Control*. Vol. 3. 1996, pp. 2490–2492.

[58] S. Mishra and M. Tomizuka. "Projection-Based Iterative Learning Control for Wafer Scanner Systems". In: *IEEE/ASME Transactions on Mechatronics* 14.3 (2009), pp. 388–393.

[59] M. Yang, G. Gu, and L. Zhu. "Parameter identification of the generalized Prandtl–Ishlinskii model for piezoelectric actuators using modified particle swarm optimization". In: *Sensors & Actuators: A. Physical* 189 (2013), pp. 254–265.

[60] V. Saveljev, S. Kim, and J. Kim. "Moiré effect in displays: a tutorial". In: *Optical Engineering* 57.3 (2018), pp. 1–17.

[61] P. L. M. Heydemann. "Determination and correction of quadrature fringe measurement errors in interferometers". In: *Appl. Opt.* 20.19 (Oct. 1981), pp. 3382–3384.