Model-Based Hysteresis Compensation and Control with 3D Printed Lousy Sensors

Dimitrios Kosmas

October 13, 2020

Summary

The field of sensor fabrication based on Additive-Manufacturing (AM) with thermoplastics, is relatively fresh and under active research. The potential of embodied 3D printed sensing is already giving rise to a variety of applications, especially in the field of soft-robotics and bio-medical engineering. Albeit, solutions on how to properly interpret the non-linear response of such sensors in control, are still sparse.

In this project, with the purpose of obtaining insight on the controllability of AM based sensors, a differential strain-gauge is designed and fabricated with the focus on addressing the symmetry of the sensor's strain gauges in terms of resistance and overall print quality by further tuning of the printing process. The fabrication consists of two materials, a flexible TPU (Ninjaflex) for the sensor's body and conductive flexible TPU (Ninjaflex) for the traces of the strain-gauges.

The sensor is placed on an experimental setup where load-unload cycles stress and compress the traces while the potential difference across each trace is measured. After post-processing the data and via a differential measurement the hysteretic response is exposed.

With aim to develop a model capable of capturing the hysteretic response of the strain-gauge, the analysis of a hysteresis model (Power-Law) is described along with several modifications based on various excitations.

Next, the inverse model is derived as means to compensate for the non-linearity. During the course of training the model under various excitations, the often observed instability of the compensator revealed the necessity to determine a stability criterion. When active, the criterion guarantees a proper compensator solution. It is then shown, given the trained excitations that the approach strongly reduces the non-linear response of the flexible 3D printed strain-gauge.

Finally, in a simulated environment, two control schemes are tested and compared. The first scheme is characterized by the direct application of the compensator upon the sensor's response. The second scheme discards the inverse model in favor of weighting the input trajectory with a forward hysteretic model. By training both compensation methods under a sine-wave excitation, it is shown that both approaches perform adequately under the trained conditions with the input-weighting performing slightly better while also being more resilient when modified excitations are considered.

Overall, the derived model and its modifications prove to sufficiently capture the sensor non-linearity. The inverse model effectively linearizes the majority of the response at a certain frequency and amplitude. The ideal simulation allowed for a low-degree of insight on the controllability, although the results hint that further investigation on the derivation process of the compensator is required.

Acknowledgements

This report describes the work towards my thesis under the MSc Systems & Control. The journey of this project is filled with enjoyment and knowledge which would not have been possible to obtain without the people surrounding it.

To Gijs Krijnen, thank you for all the active supervision, comprehensive feedback and for all the thought-provoking discussions. I am grateful for your faith in me throughout this journey.

To Martijn Schouten, thank you for all your patience, support and insight. Your help was crucial and your enthusiasm towards the project has been a welcomed bonus motivation.

To Wouter Hakvoort and Edwin Dertien, thank you for all your invaluable pointers and comments. Your feedback was vital to appropriately shape feasible project objectives.

To Alexander Dijkshoorn, Remco Sanders, Gerjan Wolterink and Leon Abelmann, thank you for all the engaging conversations and assistance during the NIFTy meetings and the time in the lab.

To Marilena and my family, without your continuous support in all aspects of my life, none of this would be possible.

Last but not least, to all my friends who made this adventure a fun and memorable time.

Dimitrios Kosmas
Enschede, October 2020

Contents

List of Figures

List of Tables

List of Abbreviations

AM Additive Manufacturing

DP Driving Points

FDM Fused Deposition Modeling

KCL Kirchhoff's Current Law

KP Krasnosel'skii-Pokrovskii

KVL Kirchhoff's Voltage Law

MEMS Micro-Electro-Mechanical Systems

NIFTy Nature Inspired Fabrication and Transduction

PEA Piezo Electric Actuator

PETG Polyethylene Terephthalate Glycol

PI Prandtl-Ishlinskii

PL Power Law

PLA Polyactic Acid

PSU Power Supply Unit

PVA Polyvinyl Alcohol

PWL Piece Wise Linear

TPU Thermoplastic Polyurethane

1 Introduction

1.1 Context

Additive Manufacturing (AM), as a technology, has been around for over 50 years but it was not until recently that it received such extensive attention, effectively leading to the emergence of new AM technologies and the rapid-growth and optimization of the existing ones [10]. This advancement gave birth to numerous solutions to the multi-material printing challenge [11]. As a consequence, printing with a variety of different materials, opens many possibilities in polymers based research. The focus of this report lies on the development of 3-D printed sensors, including the manufacturing and modelling, but with emphasis how they can be used in control loops in real-world applications.

The motivation behind this project stems from the current research of the NIFTy (Nature Inspired Fabrication and Transduction) group of the University of Twente. At the moment, a lot of focus of the group lies on exploiting AM processes as means to develop sensory systems. A key part for this research is the ability to use multi-material printing with conductive filaments, enabling the embedding of electrical circuits into rigid or soft bodies based on polymers, such as PLA, PETG to name a few.

From an application perspective, in the fields of soft-robotics, bio-mechanics, physiology and kinesiology often large strain responses are encountered, where traditional MEMS-based rigid sensors under-perform due to their low stretchability and sensitivity in such cases [12]. Although, research towards flexbile MEMS-based sensor exists in the form of miniaturizing certain sensors [12] or manufacturing based on flexible polymer films [13], these processes are still complex, challenging to manufacture and require additional assembly steps to be attached in the sensing body.

3D printing enables in-house manufacturing, higher freedom in design patterns, does not require extra assembly steps since the sensor can be embodied and is comparable cost-effective to manufacture in low-scale.

1.2 Problem Statement

AM-based sensors, despite their potential, they have been shown to exhibit a highly non-linear response [14]. Drift, hysteresis as well as creep render the task of acquiring useful feedback significantly challenging. Specifically, compensating for the hysteretic non-linearity of 3D printed sensors is a relatively fresh subject. Although the results presented in [14] show a great increase in the linearisation of the differential tactile sensor, the fact that a manual synchronization of the signals had to be used, reduced the confidence in the results.

1.3 Objectives

The main objective of this thesis is to investigate control approaches for a system equipped with a 3D printed sensor. Typically, in order to compensate for a certain behavior, the behavior itself should first be captured.

Given the fact that such sensors experience drift, hysteresis and creep, the main objective can be sub-categorized in the following tasks:

- How can the non-linearities of flexible, 3D printed sensors be captured in (a) model(s)?
- Is it possible to compensate for sensor non-linearities by means of modelling, and, if so, can this compensation be generalized?

- What control strategies can be used to successfully exploit the flexible 3D printed sensors?

1.3.1 Model objectives

Additionally, towards capturing the exhibiting behaviors it is crucial to define their role in the model. More specifically:

Hysteresis: A model should contain a memory component, carrying the information of previous states of the system. This is an integral part in order to compute the current state of the system at each instant of time for a proper representation of the *hysteretic behavior*.

Creep: The materials involved in the fabrication process of 3D printed sensor are (for the most part) polymers which are commonly modified with additions such as carbon black, in order to enhance the mechanical properties of the plastics involved or add extra properties such as electrical conductivity. It is well-known that polymer based solids experience plastic deformation when exposed to constant forces for an extended period of time. This behavior is defined as *creep*. In the words of [15], the model should (ideally) be able to capture the logarithmic time dependence of the creep relaxation dynamics. Although, in order to validate a model against such behavior, considerably long experiment should take place.

Drift: Can be described as the gradual change of the output signal independently of the measured property. In this work, the interest is primarily on the short-term drift which can be observed in relatively short experiments, and not in the long-term drift which is observed due to changes in the physical material of the sensor [].

In the following chapters we will look in the literature for previous works towards the modelling of these behaviors (mainly hysteresis). Identify the advantages and disadvantages of each model for our use-case and finally conclude to the most appropriate model using a decision table.

1.4 Report structure

There are a number of non-trivial challenges to face for a successful completion of such a project. Starting from the sensor development, setting up a proper measurement configuration, the development of a mathematical model capable of following the sensor's non-linear behavior and last, investigating different control techniques. For this reason, the current document is constructed in such a way as to treat each subject separately, taking as requirement/input the outcome of the previous work. Chapter 2 provides an overview of the often used mathematical models of hysteresis and relatable control approaches present in literature. Chapter 3 gives an overview of the sensor fabrication process. Chapter 4 describes the experiment and the post-processing of the gathered data. Chapter 5 focuses on the description and analysis of the hysteresis model as well as the derivation of the so-called compensator. A strategy based on an FEM representation of the sensor as means of digital substitution is given in Chapter 6. Possible control schemes along with tests carried out via simulations are presented in Chapter 7. Last, the overall work of the project is discussed in Chapter 8, along with several suggestions towards future work.

2 Literature

2.1 Mathematical model of Hysteresis

Attempts to model hysteretic and creep behavior have been the object of many studies since 1935 where Ferenc (Franz) Preisach first introduced his approach of creating a phenomenological model of hysteresis for ferromagnetic materials [16]. Since then, a lot of research has been focused on how to overcome the original model's shortcomings [17], like the input rate dependency, incongruency, multi-variable non-scalar inputs and the computationally heavy double integration in order to estimate the model weights.

Before an overview of the models that paved the way for the development of the model chosen for the current work, it would be beneficial to provide an overview of often used models designed to exhibit hysteretic behacior. Some common approaches are categorized in Fig. 2.1:

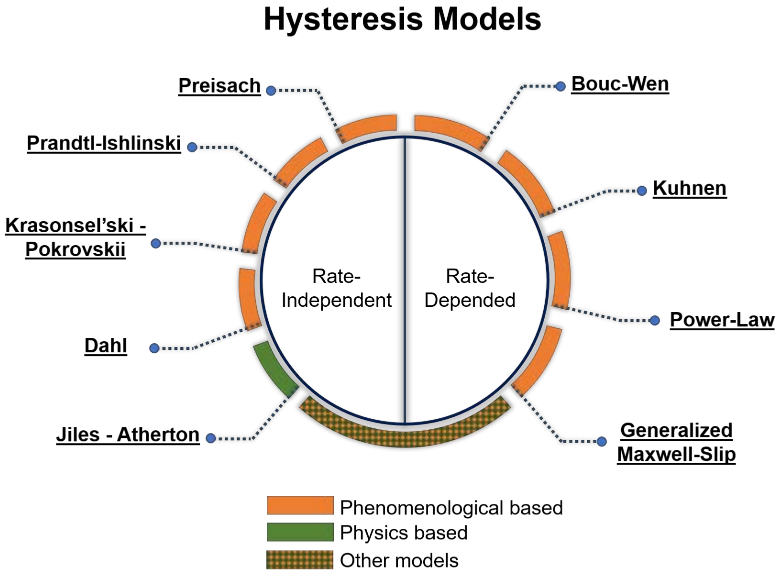


Figure 2.1: Hysteresis models categorization.

Due to the many years of research surrounding the topic, there are of-course a large number of models not mentioned or reviewed by this work. According to the literature research during this thesis, the models outlined in chapter 2 form some of the major milestones along the hysteresis understanding research.

It has been shown that 3D printed sensors exhibit hysteresis [14]. Additionally, creep phenomena are expected to be present as well due to the materials involved [18]. The latter presents a limitation to the suitable models. The creep effect, which can be informally defined as the elongation or contraction of the material due to constant forces, is basically a rate-dependency and can prove detrimental for applications where position accuracy and repeatability are important. Due to that reason, the focus is set on a rate-depended model.

In order to define a basis for the understanding, flow and terminology of this report, a brief summary of the Preisach model is required.

2.1.1 Preisach Model

Considering the phenomenological nature of the Preisach model, an approach from a mathematical perspective is in order. From the prism of mathematics, the Preisach model is comprised by an infinite set of hysteresis operators $\hat{\gamma}_{\alpha\beta}$. The input-output plot shown in Fig. 2.2 represents the rectangular shape of an elementary $\hat{\gamma}_{\alpha\beta}$ operator ($y = \hat{\gamma}_{\alpha\beta}(u)$). Here, the output can be represented with only two values, either +1 or −1, therefore, the whole model can be thought of as a two position relay with α and β the switching trigger values for each branch. From Fig. 2.2 the two branches are clear, the accending *abcde* and the descending *edfba* branch.

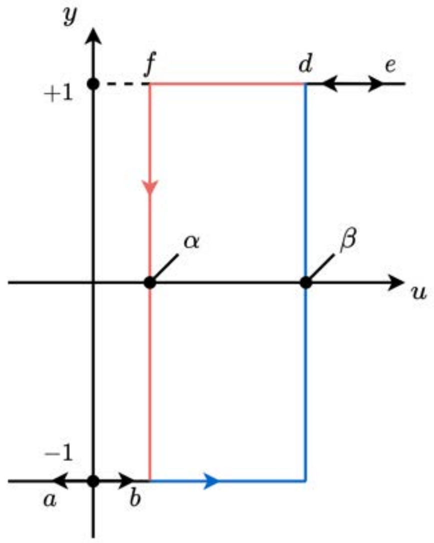


Figure 2.2: Preisach hysteresis operator input-output relationship.

In [17], where I.D. Mayergoyz gives an elaborate explanation of the Preisach model, it is shown that the $\hat{\gamma}_{\alpha\beta}$ operators are capable of representing the hysteresis non-linearities with local memories (meaning that the past values can only influence future values *through* the current output). Furthermore, the classical model is able to fit first-order transition (reversal) curves (see appendix 9.4). The model can then be expressed with the following

$$f(t) = \hat{\Gamma}u(t) = \int\int_{\alpha\geq\beta} \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}u(t)\,d\alpha\,d\beta \tag{2.1}$$

Where, $\mu(\alpha,\beta)$ is an arbitrary weight function and $\hat{\Gamma}$ a concise notation for the Preisach hysteresis operator $y = \hat{\gamma}_{\alpha\beta}$ with shape as shown in Fig. 2.2.

For an elaborate analysis of the inner workings of the model the reader is advised to read through Chapter 1 of [17], for a brief overview see Appendix 9.3. With a practical implementation in mind, Mayergoyz presents the digitization of the Preisach model (short description in Appendix 9.5) in order to avoid the heavy computation of the double integration as well as the differentiation of the experimental data in order to obtain the weight function $\mu(\alpha,\beta)$, which can further amplify the noise in the data.

2.1.2 Generalized Preisach Model

Mayergoyz, in his work towards the generalization of the Preisach model, he tackled with the following shortcomings:

- Relaxation of the congruency property.
- Ability to represent dynamical properties.
- Extend the Preisach model to work with two-input variables (as practical adaptations require [17].)

For the first, a definition of the congruency property in Preisach model, in the words of Mayergoyz in [17]:

"*All minor hysteresis loops corresponding to back and-forth variations of inputs between the same two consecutive extremum values are congruent.*"

Alternatively, the congurency property states that the loading and un-loading paths of a formed hysteresis (major or minor) loop are symmetric mirrors of each other. This effectively illustrates a limitation of the classical Preisach model when non-symmetric hysteresis loops are concerned.

Towards a model to answer these challenges the following "Non-linear Preisach" model was published in 1988 by [1]

$$\hat{\Gamma} u(t) = f(t) = \int\int_{\alpha\geq\beta} \mu(\alpha,\beta,u(t))\hat{\gamma}_{\alpha\beta} u(t)\, d\alpha\, d\beta + \int_{-\infty}^{+\infty} \nu(\alpha)\hat{\lambda_\alpha} u(t), d\alpha \tag{2.2}$$

In 2.2 it is distinct that there are two main difference in contrast to the classical model. First, the weight function μ now depends on the current input value $u(t)$, fact responsible for the *non-linear* part of the model, and second, an extra term which contains the fully reversible component of the hysteresis non-linearity [1] is added. The second term incorporates a second set of operators called the step-operators $\hat{\lambda_\alpha}$, as well as the $\nu(a)$ which, similar to the first term, is a weight function used to fit the second term to the major loop [1].

This model is capable of fitting second-order transition curves, in contrast to the original model which only allows first-order ones (Fig. 2.3).

With this model, Mayergoyz & Friedman presented an extended application range over the original Preisach model as well as greatly improved accuracy [1].

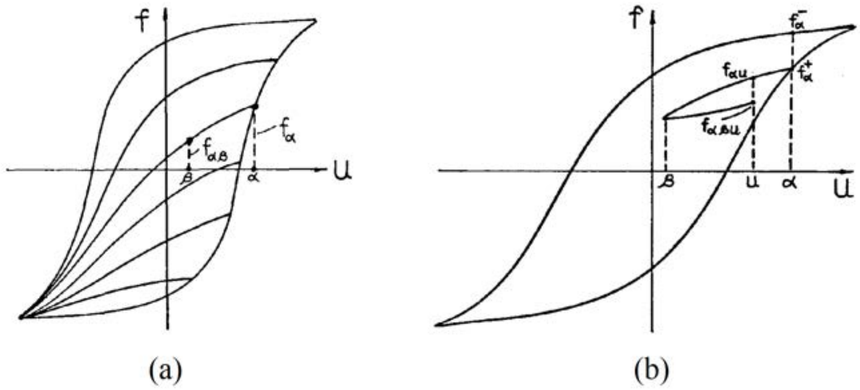


Figure 2.3: (a) First-order transition curves (original Preisach model), (b) Second-order transition curves (generalized Preisach model), as presented in [1].

As a final note towards the work of Mayergoyz in [17]. His contributions to the subject and the work towards the analysis of rate-independent hysteresis with non-local memory, has been wildly appreciated and provided the basis of understanding for a large amount of literature in that subject. For that reason it is not possible to cover all the contributions and analytical descriptions in a brief summary of his approach. Nonetheless, results and conclusions of his work are inherent in the majority of the following models of hysteresis.

2.1.3 Chua & Stromsmoe approach

The aforementioned models rely on the superposition of the hysteresis operators [17]. In 1970, Leon Chua and Keith Stromsmoe presented an alternative mathematical model able to represent dynamical hysteresis loops with local memory [2]. In this approach, the model relates the input variable $u(t)$ and the output variable $y(t)$ with

$$\frac{dy}{dt} = g(u(t) - f(y(t))) \tag{2.3}$$

Here, g and f are strictly monotically increasing functions and differentiable onto functions with non-zero slopes throughout the entire real line and satisfy $g(0) = f(0) = 0$ [2]. The $g(\cdot)$ is thought of as the dissipation function, characterizing the energy dissipation in the element. Function $f(\cdot)$ is seen as the restoring function, describing the energy storage in the element. One benefit of this approach is the relatively simple (compared to other methods at the time) determination of $g(\cdot)$ and $f(\cdot)$ function from experimental data in the case of symmetric Hysteresis loop.

Briefly, two pair of waveforms are used, one cosine waveform $y(t)$ is used for excitation while an $x(t)$ waveform is measured. Following property 9 as presented in [2] it is stated that if $y(t)$ is half-wave symmetric, then the corresponding $x(t)$ exhibits half-wave symmetry as well, then the g and f are odd functions. Expressing $x(t)$ as a sum of even functions $x_e(t)$ and odd functions $x_o(t)$ we get

$$\begin{aligned} x(t) &= x_e(t) + x_o(t) \\ x(t) &= f(y(t)) + g^{-1}(y'(t)) \end{aligned} \tag{2.4}$$

Solving the above relationship algebraically for two known points results in expressions for g and f functions. If the device in interest is not characterized with a symmetric hysteresis then more complex optimization techniques are required in order to obtain these expressions.

For the symmetric case, a step-by-step analysis of the procedure to determine the $f(\cdot)$ and $g(\cdot)$ function, the reader is advised to read section II of [2].

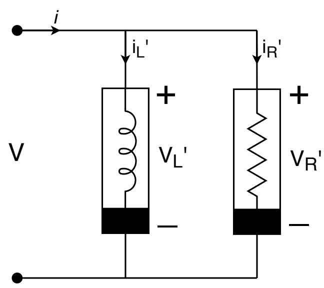


Figure 2.4: Lumped circuit model of an iron-core inductor, as presented in [2].

Fig. 2.4 shows one of the examples used in [2]. The characteristic curves of the non-linear elements of the circuit are shown in Fig. 2.5.

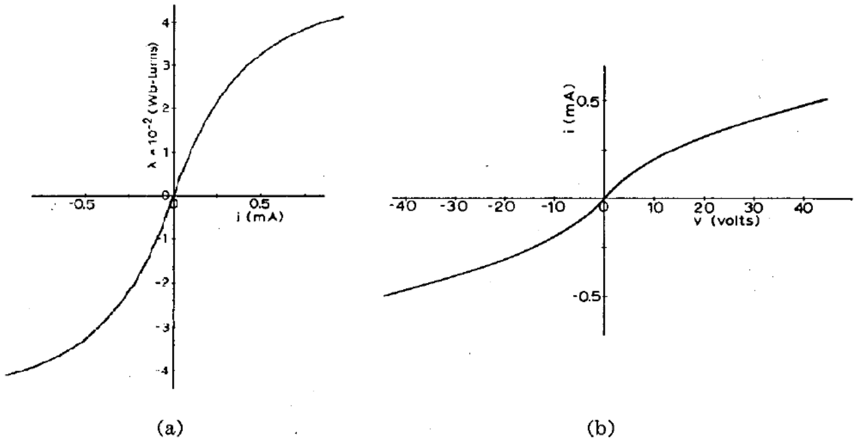


Figure 2.5: (a) $i - \lambda$ curve of the non-linear inductor,(b) $v - i$ curve of the non-linear resistor.

The model has shown the ability to form the hysteresis loop, as well as represent minor loops and increasing loop area when frequency is increased. Altough, initially developed with ferromagnetic hysteresis in mind, it is claimed, that the mathematical formulation of the model is capable of modelling stress-strain relationships in the mechanical domain. A main disadvantage, as mentioned in [2], is the inability of the model to predict dc behaviors.

2.1.4 Mechanical model alternative

S.Bobbio and G.Marrucci in 1993 presented an alternative way of modelling rate-independent hysteresis with non-local memory. In the literature, local memory property of hysteresis models is used to describe the fact that at each instant of time the output is calculated given only the current value. Non-local memory is used to describe the output values which are depended not only in the current value but on past input extrema values as well [19]. In their work, it is stated that it is important for the local memory assumption to be avoided since it was shown by [20] from an exhaustive set of experimentally obtained measurements that most materials can not be properly represented with the local memory assumption.

With first objective to reduce the computation complexity of the Preisach model, Bobbio & Marrucci introduced the idea of the combination of elements in series instead of the parallel algorithm architecture required for the computation of the Preisach model. It is shown that the model manages to sufficiently describe the hysteretic behavior.

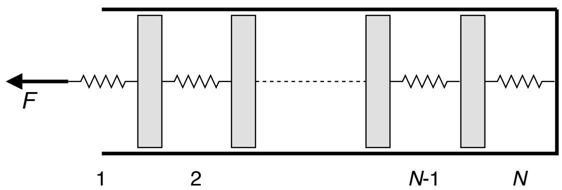


Figure 2.6: Proposed mechanical model containing N sliding pistons and N springs.

In Fig. 2.6 the mechanical system is comprised with a series of N linear springs (with assumed spring constant of 1) and $N-1$ sliding pistons. The most important assumptions for this model to work is that because of the friction, the state of each piston will not translate as long as the net force acting on it remains lower than the threshold defined as $|F_k - F_{k+1}| \leq 1$. It is further assumed that the velocity does not affect the force difference required for the motion to start [19]. The latter simplification results in the rate-independency of the model.

The known state variables of the system are the applied force F and the deformation imposed by the spring sequence $Y = \sum Y_k$. As shown in Fig. 2.7 the mechanical model is able to form hysteresis loop as well as inner loops.

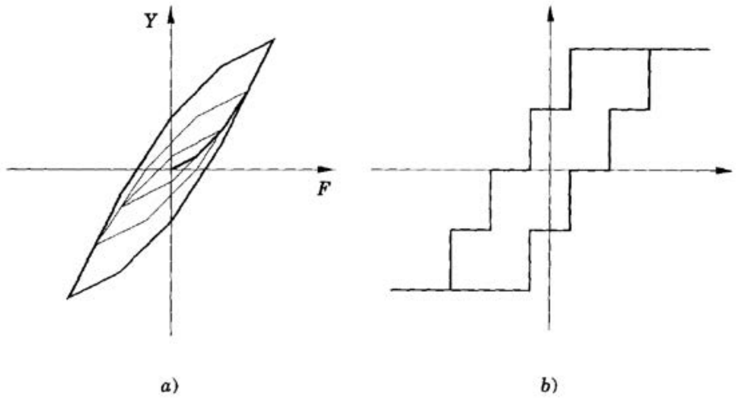


Figure 2.7: Comparison between the proposed mechanical model (a) with 4 linear springs and 3 pistons and the Preisach (b) equivalent with 4 particles.

Moreover, the model as seen in Fig. 2.6 is able to represent hysteretic behavior with non-local memory [19]. Fig. 2.8 nicely decribes the behavior of the non-local memory property. Looking at point P we can see that depending on the previous history, two different decreasing paths may occur.

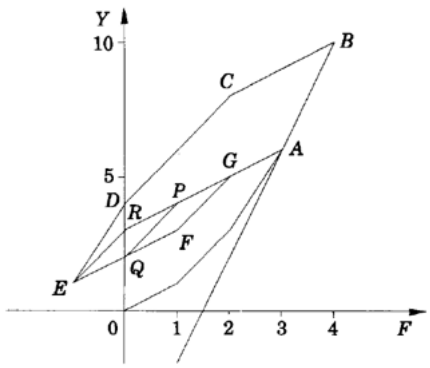


Figure 2.8: Behavior of the non-local memory.

In section 3 of [19] Bobbio & Marrucci elaborate on how these paths are generated and under which conditions the paths PQ or PR are followed.

It is crucial to point that the response of each individual element, does *not* exhibit hysteresis. The hysteretic memory is generated by the updating mechanism of the internal state which is described as

$$|U_k - U_{k+1}| \leq a_k, \quad (k = 1, 2, ..., N) \tag{2.5}$$

Here, U_k are the input forcing variables and a_k are the material constants (assumed unity throughout the paper).

Fig. 2.9 shows the model's ability to represent the major as well as the inner loops.

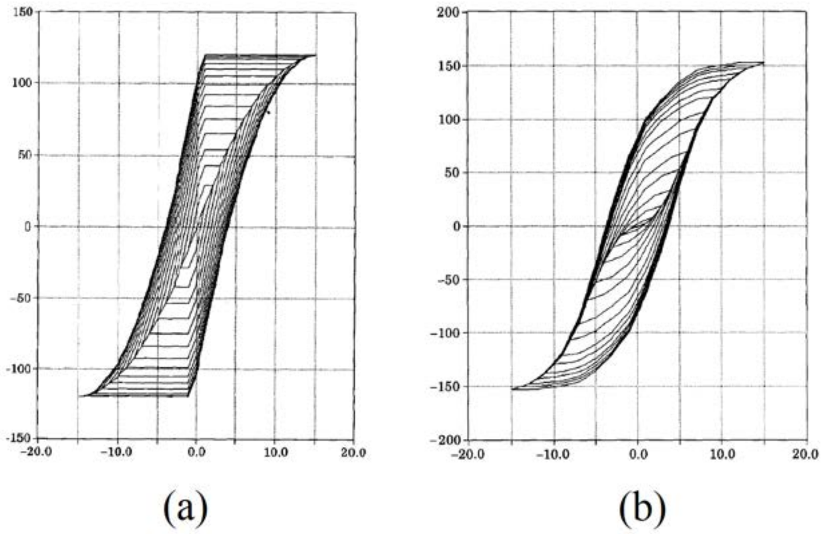


Figure 2.9: (a) Model comprised of 15 spring elements (abrupt saturation), (b) same as (a) but with four of the fifteen elements gradually saturating.

2.1.5 PWL Ladder Circuit

The works of [16], [2], [19] has led to the formulation of the model proposed by M.Parodi, M. Storace and S.Cincotti in 1994 [3]. Here, a circuit model that contains a ladder structure of linear capacitors and non-linear resistors (Fig.2.10) whose response is described with a Piece-Wise Linear (PWL) characteristic (Fig. 2.11) is proposed.

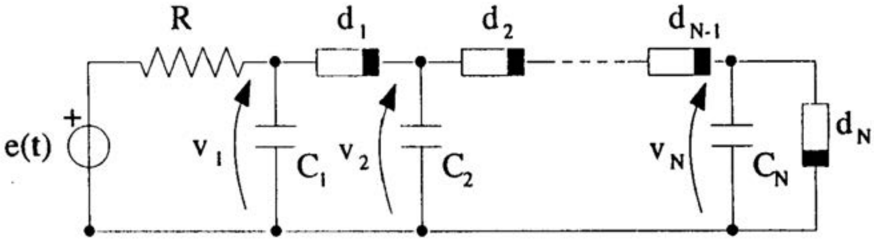


Figure 2.10: PWL ladder circuit that exhibits hysteresis.

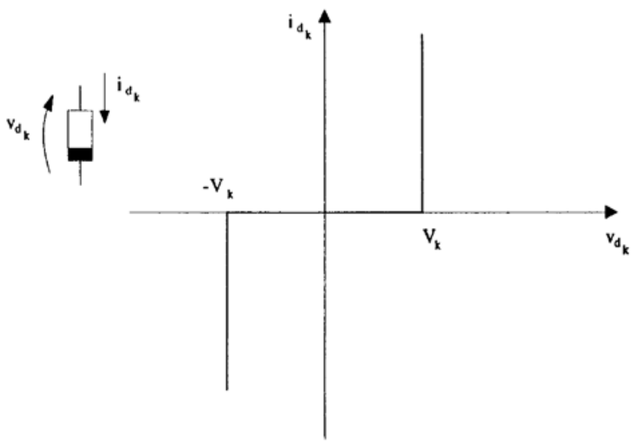


Figure 2.11: PWL characteristic of the non-linear resistors.

With the fact that an interpretation of the Preisach model in circuit terms proved to be not so direct [3], the authors attempt to transfer the model proposed by [19] from the mechanical to the electrical domain. Following the techniques of [2] the proposed lumbed circuit model was developed. In contrast though to the work of Chua & Stromsmoe, the non-linear elements of this circuit are only the resistive ones and not the inductors. The claim is that this assumption allows for the simplest form of model analysis [3].

In section 4 of their work, it is shown that non-local memory property arises from the local memory contributions of the linear capacitors [3].

Overall, the model can be considered as the result of the discretization of Preisach's model, where the N number of linear capacitors and non-linear resistors are the first parameters of this discretization [3]. The latter leads to a less cumbersome identification procedure, in comparison to the Preisach's continues model.

In their work, a comparison between the proposed PWL ladder circuit and the Preisach model is conducted, where the model's ability to maintain all the crucial elements of the hysteretic non-linearity is proved with the extension of the ability to model the non-local memory. Furthermore, a suggestion for further work towards a dynamical model of hysteresis is proposed.

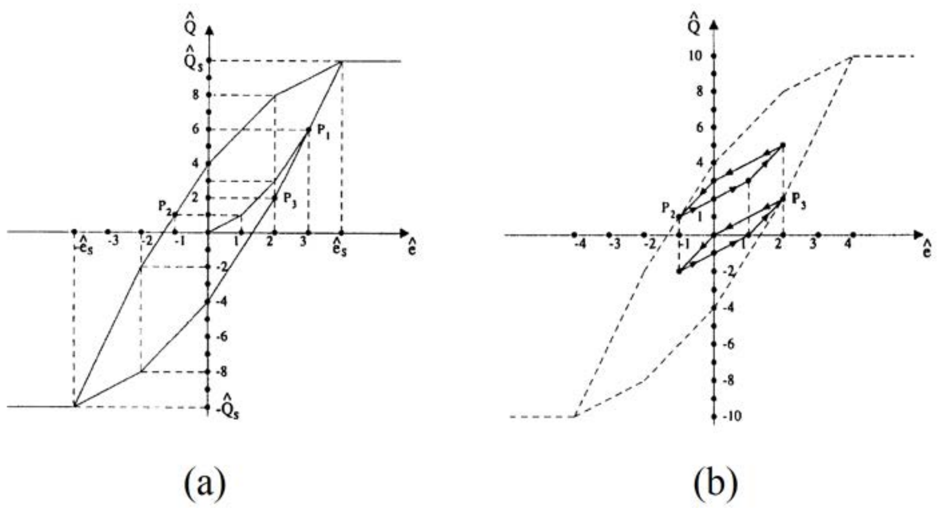


Figure 2.12: (a) Limit hysteresis cycle and 'first-polarization' curve for a circuit with N = 4, (b) two congruent minor loops. Both as presented in [3].

In 1996, the authors continue their work in [21] towards the generalization of the model. In 1998 M.Storace & M.Parodi with the work published in [22] address again the identification problem. This contribution improved the computation efficiency of the proposed model enough to make it usable with common computational packages such as MATLAB [22].

2.1.6 Kuhnen model

The increased demand in performance enhancements in nano-positioning systems, consisted commonly of Piezo-Electric Actuators (PEA's), has led to a recent focus in literature around the modelling of rate-dependent hysteresis with non-local memories. Even though, the previously described models have shown remarkable results on capturing the main characteristics of the hysteretic non-linearity, rate-dependency with "global-memory" is crucial for a complete understanding of the dynamics involved in a smart-material based system (such as the ceramics based PEA's).

K.Kuhnen, with his work as described in [23], [24], [25], [26] and [27], tackled with the challenge of identifying, modelling and compensating for hysteretic as well as creep non-linearities which are commonly observed not only in ferromagnetics but in piezoelectric, electroresistive, magnetostrictive and shape-memory alloys incorporated in smart-materials systems.

Kuhnen's approach is based on operator-based hysteresis and creep operators. More specifically, he presents a modified model based on the Krasnosel'skii-Pokrovskii (KP) [28] operator and the threshold-discrete Prandtl-Ishlinskii (PI) [29] operator. This model belongs to the family of Preisach operators, as described in the previous sections, but extended with a second set of operators which a responsible of modelling the creep behavior. As a result, the model is of purely mathematical nature as the underlying physics are not taken into account and a phenomenological approach is followed which if appropriately implemented, is able to characterize the non-linearities in a sufficiently accurate manner.

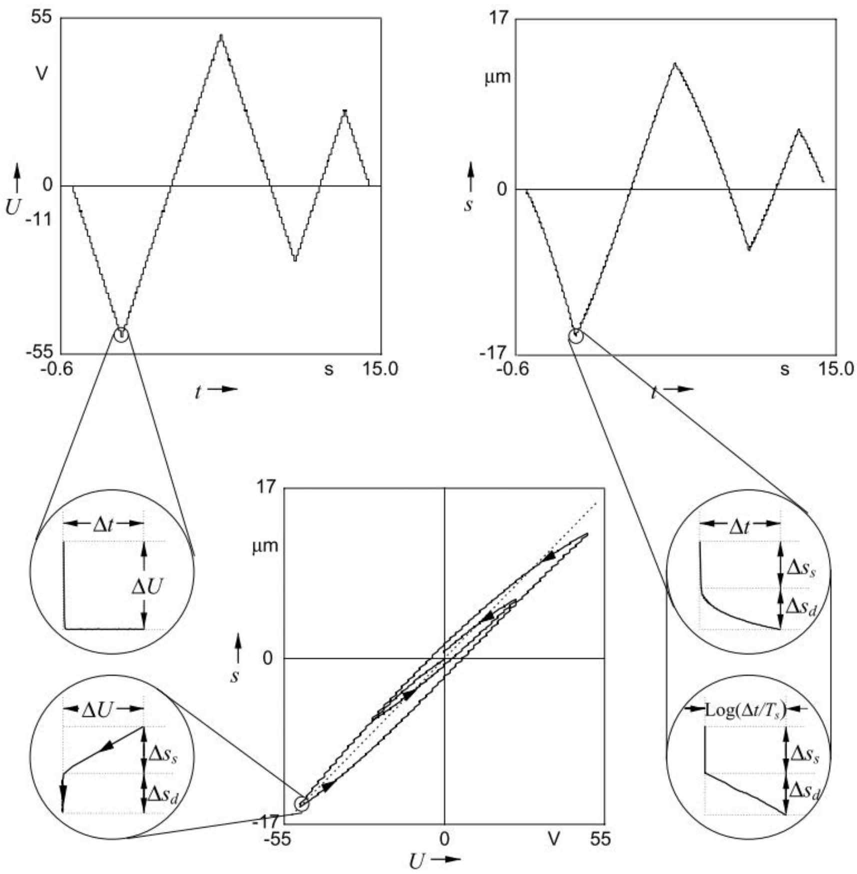


Figure 2.13: Combined complex hysteretic non-linearity with log(t)-type creep dynamics.

Fig. 2.13, at the top left, describes the response of the typical input voltage as used in micropositioning applications. At the right part, the corresponding response (displacement) of the ceramic is shown. This example aims to describe the type of the creep dynamics to be modelled. Given a time resolution of $T_s = 1$ ms the displacement response Δs consists of an immediate hysteretic part Δs_s and a time delayed creep part Δs_p of approximately the same magnitude. The local step response is shown in the bottom right encirclement against the transformed time axis $log(\Delta t / T_s)$, here it is immediate that the creep part Δs_p has an almost linear response, hence the name "$log - type(t)$ creep" [25].

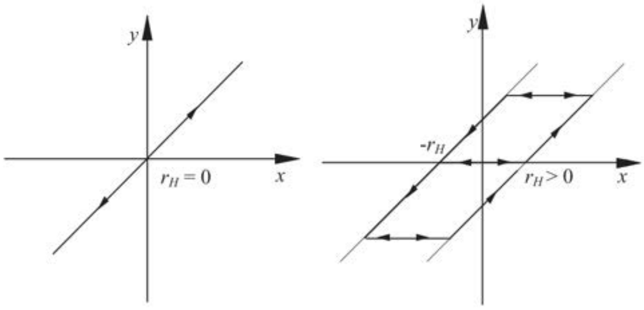


Figure 2.14: Elementary play-operator.

In summary, the two parts are modelled as follows. The complex hysteretic non-linearity is approached with a PI model:

$$\Gamma[x](t) = S[H[x]](t) \tag{2.6}$$

Where, H stands for threshold discrete Prandtl-Ishlinskii hysteresis operator defined as

$$H[x](t) = \boldsymbol{w}_H^T * \boldsymbol{H}_{r_H}[x, z_{H0}](t) \tag{2.7}$$

where $\boldsymbol{w}_H^T$ is the vector of weights, z_{H0} the vector of initial states, and $\boldsymbol{H}_{r_H}$ the so-called elementary play operator which maps

$$y(t) = \boldsymbol{H}_{r_H}[x, y_0](t) \tag{2.8}$$

In [25] defined by the recursive equation

$$y(t) = \boldsymbol{H}(x(t), y(t_i), r_H) \tag{2.9}$$

With initial condition,

$$y(t_0) = \boldsymbol{H}(x(t_0), y0, r_H) \tag{2.10}$$

Described by the skewed relay shown in Fig. 2.14

With,

$$H(x, u, r_H) = \max\{x - r_H, \min\{x + r_H, y\}\} \tag{2.11}$$

S in 2.6 stands for the PI superposition operator

$$S[x](t) = \boldsymbol{w}_S^T * \boldsymbol{S}_{r_S}[x](t) \tag{2.12}$$

where $\boldsymbol{w}_S^T$ is the vector of weights, and $\boldsymbol{S}_{r_S}$ is the vector of one-sided deadzone operators which is fully characterized by the threshold parameter $r_s \in \Re$. In [24] these parameters are explained in detail, but in a summary the S superposition operator allows the modelling of input-output trajectories which are not convex and not odd symmetric.

The resulting forms of 2.6 are shown in Fig. 2.15.

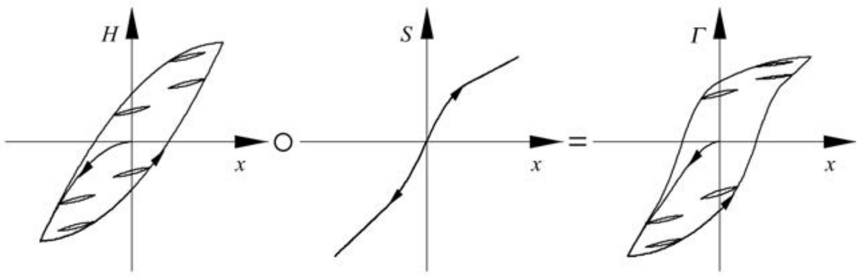


Figure 2.15: Formed loops of the asymmetrically complex hysteretic non-linearities.

For practical implementation the model can be reformed as

$$\Gamma[x](t) = \boldsymbol{w}_S^T \cdot \boldsymbol{S}_{\boldsymbol{r}s}[\boldsymbol{w}_H^T \cdot H[x, \boldsymbol{z}_{H0}]](t) \tag{2.13}$$

Here, $\boldsymbol{w}$ are the weights and $\boldsymbol{z}_{H0}$ the initial conditions.

The creep operator is modelled as a second PI operator

$$K[x](t) = \boldsymbol{w}_K^T \cdot \boldsymbol{K}_{\boldsymbol{r}_k \boldsymbol{a}_k}[x, \boldsymbol{Z}_{K0}](t) \cdot i \tag{2.14}$$

Finally, the combined model is formed as

$$\begin{aligned} \Gamma[x](t) &= S[H[x]](t) + K[x](t) \\ &= \boldsymbol{w}_S^T \cdot \boldsymbol{S}_{\boldsymbol{r}s}[\boldsymbol{w}_H^T \cdot H[x, \boldsymbol{z}_{H0}]](t) + \boldsymbol{w}_K^T \cdot \boldsymbol{K}_{\boldsymbol{r}_k \boldsymbol{a}_k}[x, \boldsymbol{Z}_{K0}](t) \cdot i \end{aligned} \tag{2.15}$$

In equation 2.15 the extended PI model proposed by Kuhnen is shown.

Due to the main objective of K.Kuhnen's work being the compensation and control of systems that exhibit the aforementioned complex non-linearities, his work is re-visited from a control point of view during the literature study surrounding the topic of control.

2.1.7 Power Law PL model of hysteresis and creep

Along the same motivation as K.Kuhnen (to capture the rate-depended phenomena typically observed in PEAs), the previous authors M.Parodi and M.Storace together with M. Biggio, A. Oliveri & F.Stellino [15] proposed a modified and extended circuit model, able to capture the rate-depended hysteresis as well as the "$log - type(t)$" creep dynamics.

Based on the work of [3], [19] and [21] the new circuit (2.16) is formed as the combination of two distinct sub-circuits. In a summary, sub-circuit (a) is responsible of modelling the hysteretic memory through a parallel architecture comprised with a linear capacitor C and a non-linear resistor R whose characteristic is as shown in 2.11. The second sub-circuit, contains the parameters related to the identification procedure, namely a voltage bias V_b, a PWL non-linear resistor and a set of in-series connected VCVS's (Voltage Controlled Voltage Source's [15]). The VCVS's are configured by the weighed voltage across the linear capacitor of sub-circuit (a).

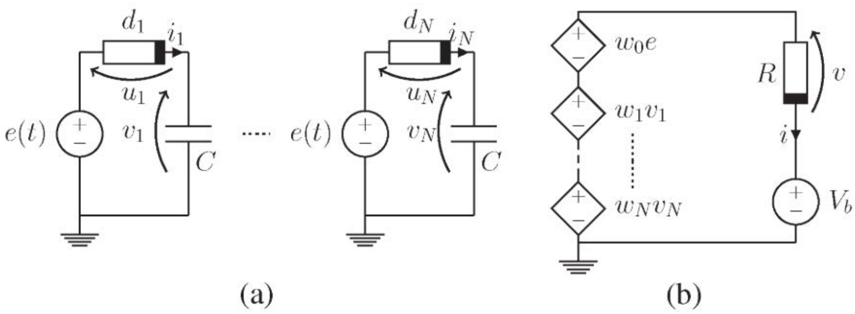


Figure 2.16: Circuit model for Hysteresis and Creep.

In terms of the original PWL ladder network model, it is shown that the new apporach based on the parallel structure is able to exhibit all the previously described attributes. Apart from the updated circuit representation, a new representation of the non-linear resistive elements of the circuit (Fig. 2.17 (b)) extends the model with the ability to capture the "$log - type(t)$" creep relaxition dynamics [15]. The novel feature, is the representation of the middle part of the PWL linear function of the resistive elements with the odd power function shown in equation 2.16. The thresholds, as described in Fig. 2.17 (b), define the Driving Points (DP). As it is shown in [15], depending on the configuration of the DP's the model is capable of capturing the logarithimic time dependence of the creep dynamics.

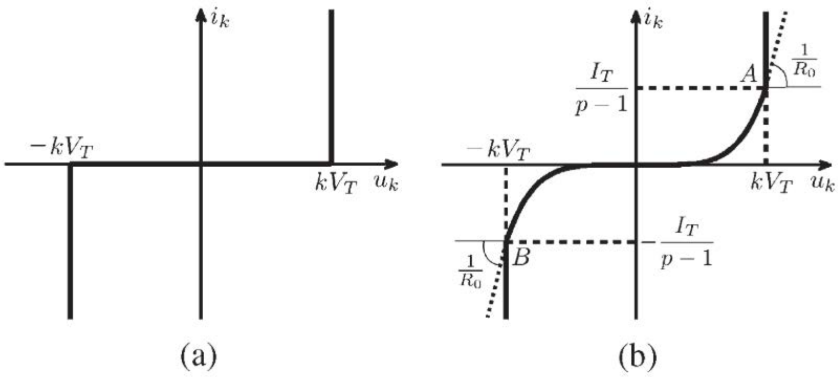


Figure 2.17: (a) PWL linear characteristic, (b) PL adapted characteristic.

$$i_k = \frac{I_T}{p-1}\left(\frac{u_k}{kV_t}\right)^p, \quad p = \text{odd} \tag{2.16}$$

It is assumed that the non-linear resistor of the second sub-circuit can be represented as the sum of weighted PWL linear functions $\phi(t)$

$$f(i) = \sum_{k=1}^{M} a_j \phi_j(t) \tag{2.17}$$

In the system as whole, the input is the $e(t)$ voltage source of sub-circuit (a) and that total output is the current $i(t)$ running through the non-linear resistor of sub-circuit (b). The constitutive relation of the latter is described as $u = f(i)$ with $u(t)$ (eqn. 2.18) the voltage across the resistor.

$$u(t) = w_0 e + \sum_{k=1}^{N} w_k u_k - V_\mathrm{b} \tag{2.18}$$

With, w_0, w_k weights and u_k the voltages across each capacitor (described by equation (6) of [15].

Finally, the entire mode output is computed with the inverse $i(t) = f^{-1}(u)$.

The Power Law model is shown to exhibit both the hysteretic and creep non-linearities.

In [4], the PL model is implemented in an open-loop control scheme for hysteresis and creep compensation and in [5] an extended comparison between the last two models (Kuhnen and PL) is conducted. Although, the discussion and results of these papers are better suited for the literature towards the control strategies that will follow.

2.1.8 Results

With the brief summary of some commonly used hysteresis models, we can formulate a decision table (2.1) to help us derive to a conclusion regarding the modelling approach to follow.

Table 2.1: Hysteresis modeling technique decision table.

	Rate-independent	Rate-dependent	Local-memory	Non-local-memory	Minor loops
Classical Preisach	Yes	No	Yes*	Yes	Yes
Generalized Preisach	Yes	No	Yes	Yes	Yes
Chua & Stromsmoe	Yes	Yes	Yes	No	Yes
Bobbio & Marrucci	Yes	No	Yes	Yes	Yes
PWL ladder	Yes	No	Yes	Yes	Yes
Kuhnen	Yes	Yes	Yes	Yes	Yes
Power Law	Yes	Yes	Yes	Yes	Yes

1

Looking at the objectives set for our case (section 2.1), the above tables tells us that only the *Kuhnen* and *Power Law* model have the capability to capture the creep dynamics. In order to choose between the two, a MATLAB toolbox, provided by [30] is to be used. This toolbox, incorporates four different hysteresis modeling techniques (including the latter two models). Therefore, in order to obtain a final conclusion about which model to use, first some experimental data from a 3-D printed sensor will be obtained and used with the toolbox facilities to get an estimate of the performance of each model.

Another advantage that may prove detrimental for the selection of the proper technique, is the computation efficiency of the model. In [5], where the comparison between these models is conducted, it is claimed that the Power Law model, achieves the accuracy of the Kuhnen model by employing a lower number of parameters.

[1] Preisach can generally model hysteresis with local memories, but that does not mean that all hysteresis non-linearities with local memories can be modelled with Preisach. There are in fact some hysteresis non-linearities with local memory that has been shown that they cannot be modelled with the Preisach model, like the hysteresis observed in single Stoner-Wohlfarth magnetic particles [17].

2.2 Control strategies towards hysteresis compensation

Towards compensation of the hysteretic non-linearity, literature focusing directly on 3D printed sensor non-linearities is non-existent as per research realized at the time of the current assignment. Although, as mentioned, the subject of the hysteresis compensation is heavily investigated in the past few decades, especially towards nano-positioning control for Piezo-Electric Actuators (PEAs).

There are several different approaches for compensating PEA hysteresis and creep where the majority of them is based on feedback loops based on bulky sensors [31], or current control methods instead of voltage control that appear to exchibit a much less hysteretic response in the expense of complicated controllers and electronics [32]. Although, none of this approaches is applicable in our sensor scenario since the use of extra sensor to compensate for already existing 3D printed sensor renders the concept unnecessary and any kind of current control is not applicable since the sensor is not actively actuated. The third option of compensating for these non-linearities in PEAs is based on the modelling of such behavior and deriving the inverse as means to eliminate them. The latter is a sensorless open-loop approach whose performance is heavily depended on the model accuracy.

Models as the ones of Kuhnen and Power-Law are specifically developed with that concept in mind. Although, as shown in the previous section, the model and inverse generation for both models is substantially different, the control implementation of both is quite similar where the inverse model (compensator) is directly applied prior to the plant (PEA).

If the PEA is considered as single input-output block as follows:

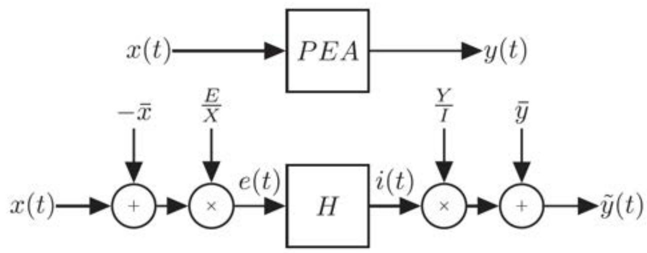
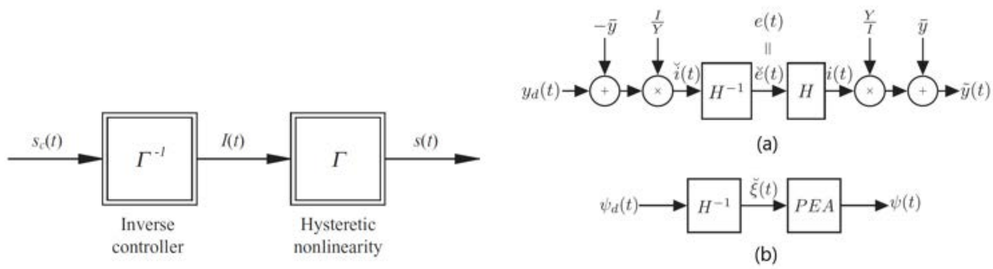


Figure 2.18: Non-compensate PEA system. Actual system (top), modelled equivalent (bottom) [4]

Then, the compensation scheme is described as:



(a) [Kuhnen,2003 [25]] **(b)** (a) Model, (b) actuation test scheme [4]

Figure 2.19: Typical Open-Loop hysteresis compensation scheme for PEAs.

The models Γ and H are described in the previous section for both the Kuhnen and Power Law model respectively.

Kuhnen derives the controller by taking the inverse of 2.15 which is expressed as:

$$\Gamma^{-1}[y](t) = \boldsymbol{w}'^T_S \cdot \boldsymbol{H}'_{\boldsymbol{r}s}[\boldsymbol{w}'^T_H \cdot \boldsymbol{S}_{\boldsymbol{r}'_s}[y], z'_{H0}](t) \tag{2.19}$$

Where for the Power-Law the inverse is derived from 2.18 such as the compensated output is represented with:

$$\breve{e}(t) = \frac{\breve{u}}{w_0} + \frac{V_\mathrm{b}}{w_0} - \frac{1}{w_0}\sum_{k=1}^{N} w_k \breve{u}_k \tag{2.20}$$

Initialization

It is important to note here that in order for 2.20 to work as the inverse of the H a proper initialization is required. In [4] this is done by an input wave-form that synchronizes the forward and inverse model to the same conditions. Due to the hysteretic non-linearity the initial conditions of the state of a PEA at rest are not always known. The same intuition can also be applied in the sensor non-linearity. Therefore, the implementation of a robust initialization procedure is crucial. Different techniques have been introduced to tackle with this challenge, such as an excitation with a decaying sine [33] or a cyclic excitation of a triangular signal [34]. In [4], an alternative approach is proposed by acting on the voltage $e(t)$ (Fig. 2.19(b)) such that:

$$e(t) = e_\mathrm{sync}(t) = \begin{cases} E_0, & t < t_0 \\ E_0 - 2NV_\mathrm{T}, & t \geq t_0 \end{cases} \tag{2.21}$$

Where N is the cell number and V_T the voltage threshold defined as $V_\mathrm{T} = E/(N+1)$.

With $NV_\mathrm{T} \leq E_0 \leq E$ it is shown in [4] that $\forall u_k(t_0^-), \forall \breve{u}_k(t_0^-)$ it holds that $u_k(t_0^+) = \breve{u}_k(t_0^+) = E_0 - (2N-k)V_\mathrm{T}$, with $k = 1,..,N$. Additionally,

$$e(t) = e_\mathrm{sync}(t) = \begin{cases} E_0, & t < t_0 \\ E_0 + 2NV_\mathrm{T}, & t \geq t_0 \end{cases} \tag{2.22}$$

With $-E \leq E_0 \leq NV_\mathrm{T}$ it is again shown that $\forall u_k(t_0^-), \forall \breve{u}_k(t_0^-)$ it holds that $u_k(t_0^+) = \breve{u}_k(t_0^+) = E_0 + (2N-k)V_\mathrm{T}$, with $k = 1,..,N$. Meaning that since $u_k(t_0^+) = \breve{u}_k(t_0^+)$, then $\tilde{y}(t) = y(t)$ for $t > t_0$.

Identification

Both of the latter model-based techniques require measurement data so that the weights are estimated towards the best fit possible. Subsequently, last step before applying the compensator is the identification process.

Kuhnen applies a quadratic optimization problem using the generalized-error function as minimization function (see eqn.33 of [25]) upon the normalized data. Towards the identification for the Power-Law model, the authors use the combination of an exhaustive search for the parameters N, p, M and S [4], again upon the normalized data. The remaining weights are obtained by minimizing the contrained quadratic cost function as defined in eqn. 11 of [15].

Since an analytical description of the identification process is not the main objective of the current work, emphasis will be put upon the derivation of the model and its inverse where towards the identification problem, widely available tools such as the Optimization Toolbox from [35] may be used as means to obtain the estimated parameters/weights in an off-line processing manner.

Results & Comparsion

A qualitative comparison between the two latter described inverse-model compensation methods is discussed in [5]. The authors (the team behind the development of Power-Law model) performs detailed comparison of the two models in question by first describing the basis of comparison in terms of cells (PL)/hysteresis operators (Kuhnen). This results in a qualitative comparison between the two, with similar variables to be estimated in each case.

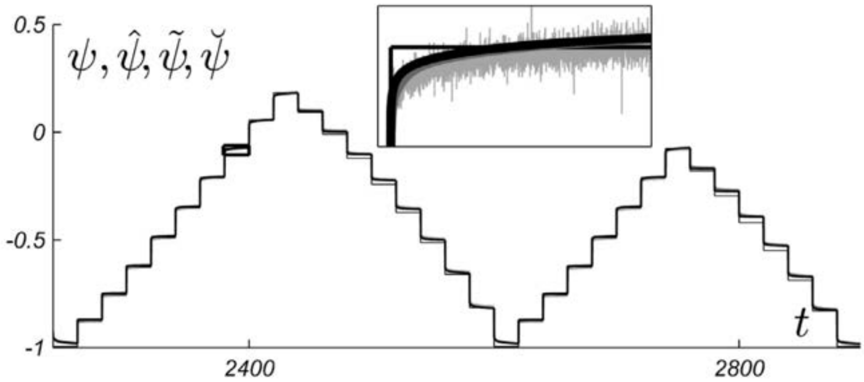


Figure 2.20: Time evolution of the PEA output ψ (light greyline), Kuhnen's model output $\hat{\psi}$ (dark greyline), power-law modeloutput $\tilde{\psi}$ (black thickline), Preisach model output $\breve{\psi}$ (black thinline) [5].

Table 2.2: Fitting errors for a given set for both PL and Kuhnen models [5]

	Power-Law	Kuhnen
Training set		
min RMSE	$5.7\ 10^{-3}$	$5.6\ 10^{-3}$
mean RMSE	$6.0\ 10^{-3}$	$7.1\ 10^{-3}$
variance RMSE	$1.15\ 10^{-7}$	$9.49\ 10^{-7}$
Test set		
min RMSE	$6.9\ 10^{-3}$	$6.5\ 10^{-3}$
mean RMSE	$8.2\ 10^{-3}$	$12.3\ 10^{-3}$
variance RMSE	$7.78\ 10^{-7}$	$1.73\ 10^{-5}$

The fitting results of table 2.2 provide enough confidence for a good fit towards a hysteretic response. Additionally, from Fig. 2.20 the fitting results of both models along with the original Preisach one is plotted against the real data of the experiment. It is imminent that the Preisach model shows a considerable lower performance which is due to rate-independence characteristic of the model where the creep phenomena present in PEAs are not captured [5].

2.2.1 Conclusion

Direct application of the inverse model prior to the hysteretic plant was shown in literature to be a reliable method as long as the model sufficiently captures all the non-linearities. Of course, these results demonstrate compensation towards PEA whereas the current work focuses on sensors that are passively actuated from a given plant. However the same concept can act as the initial goal for compensation where the compensator is directly applied, but in this case after the sensor. More on the adaptation towards 3D-printed sensors can be found in chapter 7.

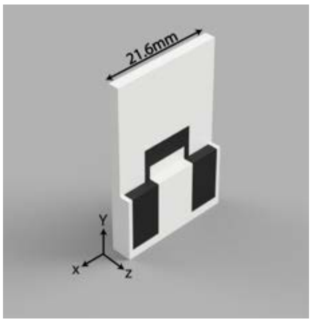
3 Sensor Development

3.1 Introduction

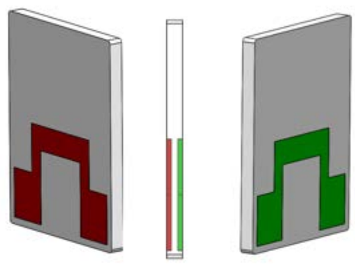
A first step towards the characterization of the non-linearities of 3D printed sensors would be to fabricate a number of sample sensors to be used in gathering experimental data. In this chapter the design and fabrication procedure is discussed. Additionally, an experiment towards the resistance-drop effect of fresh printed sensors is presented as means to identify the minimum *resting* period until the resistance settles to a certain value, after which the sensor can then be used with (more) confidence.

3.2 3D Model

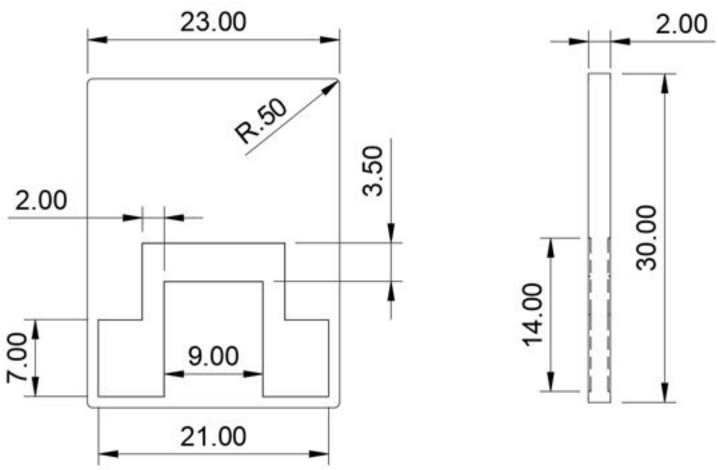
Following on the basis of the design of [14] a new 3D model is generated. As described in the introduction chapter, the target is to improve the performance of the previous sensor such that the differential measurement potential for control is maximally exploited.



(a) 3D model of the tactile sensor as presented in [14]



(b) 3D model of the modified tactile sensor. Bottom R_1 (red) and top R_2 (green).



(c) CAD drawing of the modified tactile sensor. Dimentions in [mm].

Figure 3.1: CAD models of previous and current work.

In the work of [14], a very simple strain sensitive pattern with two traces (top & bottom) was designed and fabricated (Fig. 3.1a). The goal is to keep this simple design, while improving on a couple of shortcomings. First, the sensor symmetry issue, meaning that the magnitude of the top and bottom trace resistances need to be as similar as possible, Secondly, to configure the fabrication process such that the filament dripping/mixing is eliminated. These improvements are crucial for the final goal which is to use a sensor for a position control scheme since the response measurement of the sensor is the determining value fed into the controller. Section 3.4 elaborates on the proposed improvements.

3.3 Equipment

The main equipment available to perform the design and fabrication task consists of:

- Diabase H-series 3D printer (see Fig. 3.2).

  This multi-material printer (see [36] for review towards multi-material FDM printing) is capable of simultaneously printing with up to 5 filaments. The filament switching mechanism is based on a turret configuration that places the idle nozzles at a safe distance from the building platform. For the current purpose 3 out of 5 filament paths are used.
- PI-ETPU 85-700+ conductive flexible filament [37].
- Ninjatek EEL TPU conductive filament [38].
- Ninjatek TPU flexible filament [39].
- E-Sun PVA soluble support filament [40].

Software:

- Dassault Solidworks, as 3D modelling software.
- Simplify3D, as slicer software.

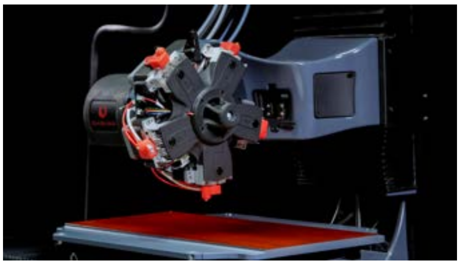


Figure 3.2: Diabase H-series Multi-Material 3D printer.

3.4 Improvements

A Main issue with the previous sensor to address is the asymmetry of the sensor's gauge resistances due to the printing process. In a typical FDM manufacturing manner, the sensor in [14] has been printed with its back side flat on the printer's heated bed (see coordinate frame in Fig. 3.1a). Typically, the initial layer is squeezed on to the heated bed, as means to provide consistent adhesion. Although, in the case of the current design, one of the two resistive traces lies on the back side as well. Subsequently, this trace's conductive lines end up forming in a more condense manner in contrast to the top resistive trace. This is indeed exemplified by the measurements in [14].

The idea to solve this issue is to raise the sensor's first layer above the heated bed. A number of approaches can be used to achieve this. For example: including the so-called *raft*, which is basically a number of 3D printed layers that act as solid base support which is to be removed during post-processing. With a raft the risk of a rough first layer of the sensor is introduced, due to the detachment step, or even blending of the support material with the model material.

Table 3.1: R per strain-gauge

Sensor	Material	R_1 (top) [kΩ]	R_2 (bottom) [kΩ]	$2\frac{R_1-R_2}{R_1+R_2}$ [%]
Sample 0 [14]	PI-ETPU	1.58	2.78	-55
Sample 1	EEL TPU	15	16.82	-11
Sample 2	PI-ETPU	12.7	28.2	-75
Sample 3	PI-ETPU	14.3	35.8	-86
Sample 4	PI-ETPU	15.7	22.4	-35

Another solution is to use water-soluble support material (e.g PVA) in order to avoid damage to the sensor's first layer during post-processing. With that approach, the 3D printed sensor is placed into a water bath for a given amount of time until the soluble material completely dissolves. While practicing this technique, it was noted that it was challenging to remove all the PVA material from the bottom side, some residue was always present.

To address this issue, the so-called *ironing* feature is used (first introduced by Ultimaker Cura slicer engine [41]). The idea is to use PVA material for the few first layers as means to avoid the squeezing of the bottom resistive trace, and then via *ironing*, smooth out the layer upon which the sensor printing process begins. Finally, this approach provided improvements towards the sensor's symmetry which is demonstrated by the resistance (R) values as shown in table 3.1 (samples 1 & 4).

Some comments for table 3.1. Sensor 0 is fabricated > 12 months before the shown measurement which (as shown in section 3.5.1) has an impact on the trace resistance. Additionally, samples 2 & 3 reveal a high-asymmetry where the bottom gauge has $\approx x2$ resistance value. This is due to the PVA raft attempt (more on sec. 3.5) without ironing feature, where residuals of the PVA form a thin layer over the conductive trace.

The choice of the number of conductive layers is arbitrary. Although, in order to provide a stronger bond and path for the current to flow, at least two conductive layers per strain-gauge are suggested.

3.5 Fabrication

With the aforementioned equipment and insights, a total of 4 sample sensors are printed. Table 3.2 summarizes the printing parameters and 3.1 describes the resistance values per sensor per strain-gauge. In Fig. 3.3b the PVA raft is shown, which is afterwards removed during the post-processing.

(a) PVA support shown in yellow.

(b) 3D printed tactile sensor sample 4 on top of PVA *ironed* layer.

Figure 3.3: PVA raft addition as means to raise first layer from the building platform.

Table 3.2: Fabrication top-level parameters

Parameter	Value
Layer thickness	200 µm
First layer thickness	150 %
Hotend temperature	
Ninjatek EEL	200 °C
Ninjatek TPU	200 °C
PI-ETPU	200 °C
Heatbed temperature	60 °C
Printing speed	35 mm s^{-1}
Support type	custom PVA raft

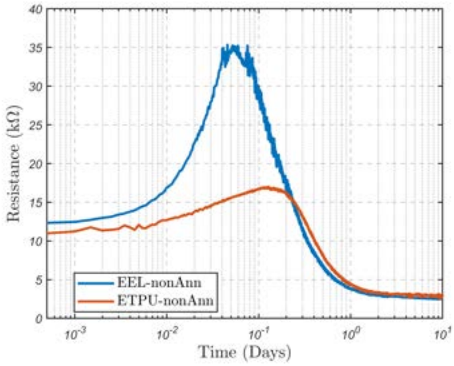
Newly printed conductive traces seem to show a decrease of their resistance value over time (more on this in section 3.5.1). As observed, after allowing a settling time as described in the following section, the resistance from each trace is significantly reduced and stabilized.

3.5.1 Resistance drop effect

Through past fabrications of 3D printed sensors within the NIFTy group, it has been observed that a significant drop in the resistance of the traces of 3D printed conductive structures takes place after the printing process. This effect is shared across most of the commonly used generic conductive filaments. In an attempt to obtain a better practical insight regarding the usability of sensors employing such materials, an extended measurement experiment has been carried out.

A total of 4 samples, blocks of purely conductive filament and size 30 mm × 10 mm × 1 mm, were printed. A pair of Ninjatek EEL conductive filament and a pair of PI-ETPU filament. One set (1x EEL, 1x ETPU) went through heat treatment over a period of 16 h at 150 °C. The printing process of the second pair was planned to complete simultaneously with the heat treatment (annealing) of the first pair.

A custom clamp was manufactured with 4 screws along its length in equal distances. In the inside of the clamp a total of 8 conductive copper tape strips (3M 1181, 6 mm wide) have been placed (2 per block) and 4 wires per sensor were soldered as means to obtain 4-point measurements. The clamp is then connected to a multimeter with a 10-channel multiplexer (Keithley 2000 with 7011 multiplexer card). The multimeter is interfacing with MATLAB through serial communication [42]. A control script is set to obtain 1 measurement per minute per sensor and log the data. Fig. 3.4 shows the experimental setup and following are the results of this experiment.



(a) Non-annealed pair. **(b)** Annealed pair.

Figure 3.5: Resistance drop effect experiment results.

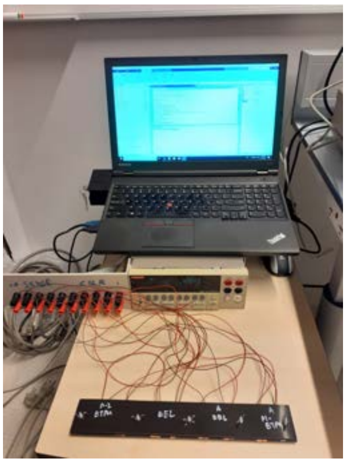


Figure 3.4: Lab setup for resistance-over-time measurement.

From Fig. 3.5 it is immediately clear that heat treatment greatly reduced the resistance (~$x10$ for EEL and ~$x5$ for ETPU) as well as provided a more stable behavior. It also interesting, that non-annealed samples (Fig. 3.5a) are showing a steep increase during the first hours and then a significant drop until saturation at $\approx 2-3$ days This behavior is in line with previous observations of the group. Moreover, due to the known creep effects of the polymers in use, it may be possible that the peaking present at the non-annealed samples is the effect of the creep relaxation dynamics due to the constant force acting from the clamp screws, although this is just a conjecture. In such a case, the annealed sample also shows a much better response where no initial peaking is observed. Furthermore, it can be deducted from Fig. 3.5a, that allowing a settling time of 2 days after the print process is finalized, will suffice in order to obtain more consistent results later on. This is best observed when plotting the results versus the logarithm of the time.

As a last note, the EEL filaments were both average quality samples, the reason being the brittle behavior of the printed part, possibly due to the long exposure of the filament spool in humidity, although this is just a guess. This fact, may also provide a reason for the "unstable" measurements seen in 3.5b.

Following on, samples 1 & 4 will be used for various measurement sessions (see chapter 5), due to the better printing results of these sensors and for a side comparison between the two distinct conductive filaments.

3.6 Conclusion

Four sensor samples are fabricated where the original issues observed in [14] are addressed. The trace resistances are of the same magnitude although there is room for further improvements in that area, as the resistance values suggest. The new print settings produced a clean sensor in terms of filament mixing during printing. Lastly, following the resistance-drop experiment a minimum resting period of 2 to 3 days is mandatory before safely measuring and implementing a measuring circuit for the non-annealed strain gauges.

4 Experimental Setup and Measurements

4.1 Introduction

In this chapter, the setup for gathering the required data is discussed in terms of equipment configuration, overview of an experiment based on a sine-wave excitation, along with the analysis and post-processing of the resulting data. The importance of the sensor symmetry along with the hysteretic nature of the response of the 3D printed sensor is revealed and, lastly, a description of the work of [9] is made, where experiments with various excitations (decaying sine, frequency sweep and noisy signal) are carried out as means to further evaluate the derived model.

4.2 Components

With the fabricated sensor, the next task is to develop a setup to serve as experiment and measurement platform. The main components for, simple, sensor-focused, isolated measurements are enumerated:

1. Fabricated strain sensor.
2. Oscilloscope:
   (a) Pico Technologies Picoscope 5443D.
   (b) TiePie Handyscope HS5.
3. SMAC linear actuator LCA25-050-15F.
4. SMAC controller.
5. PSU 1-30V.
6. PC - MATLAB.
7. Mechanical assembly.

The main purpose of the setup is to measure voltage changes in both traces of the sensor under various excitations. Towards this goal each component, along with the related configuration, should be described in further detail, with an eye to reproducibility.

The sensor and the fabrication have already been described in chapter 3. Powering of the experiment is done by a generic 1-30V PSU.

4.2.1 Oscilloscope

Figure 4.1: 4-Channel PicoScope 5443D.

Figure 4.2: 2-Channel Handyscope HS5 module.

For data acquisition, a digital oscilloscope is used in combination with the accompanied software. Initially the Picoscope 5443D (Fig. 4.1) is employed, as part of the first measurement with a sine-wave excitation (more details in section 4.3.1). The scope has a maximum resolution of 16-bit, but in case all 4-channels are measured simultaneously the maximum resolution is of 14-bit.

From the work of [14] a challenge on the synchronization of the signals is considered as a possible source of measurement inaccuracy. In order to address this issue, one solution would be to measure all the interesting signals under the same device, effectively generating measurements synchronized under the same time scale.

In order to achieve this, the encoder signal (direction from SMAC actuator to SMAC controller) needs to be extracted and measured by the employed data acquisition device. The sensor used inside the SMAC linear actuator is a differential quadrature encoder with 5 µm resolution. Typically, in order to take full advantage of this hardware, five measurements would be required for the encoder alone. As described by Fig. 4.3, the two encoder channels incorporate differential measurements ($A+, A-, B+, B-$) and the encoder Z channel, which provides a single pulse when the stroke is aligned with the default home position, which is identified during startup in the initial calibration cycle.

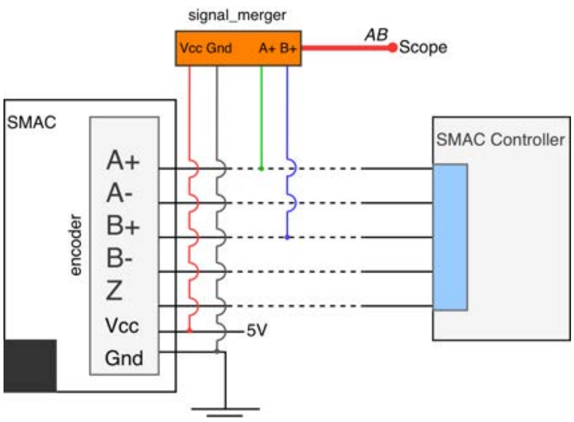
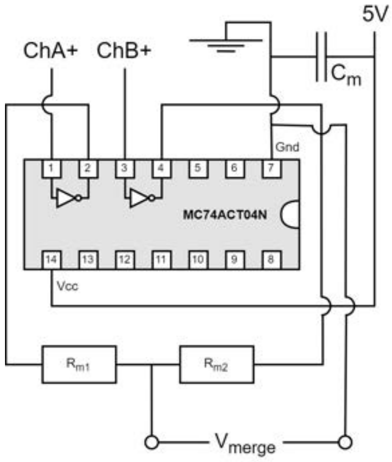


Figure 4.3: Encoder signal extraction diagram.

In the current situation the setup is limited to one data acquisition device, hence measurements by 4-channels simultaneously is the maximum possible. Due to this fact, a compromise needs to be made. The most basic form of encoder reading is by reading the single end channels A and B, omitting that way the differential channels as well as the zero channel. Furthermore, the two channels are combined into one merged signal in order to reserve one more channel for a possible reference voltage measurement as means to account for potential power-supply output voltage instability.

(a) Encoder merger cable.



(b) Merger schematic.

Figure 4.4: Merger cable for encoder channels measurement.

The *merger*, as the word suggest, serves as a signal merger for the encoder signals A^+ and B^+. It employs a hex inverter buffer as the signal amplifier [43]. A voltage divider with resistance values of R_{d1} =1 kΩ R_{d2} =2 kΩ is used to assign a unique voltage magnitude for each encoder channel. That way the combined signal is described by 4 different voltage states. These states are used during post-processing in order to invert the merging, decompose the combined signal into two distinct signals and reconstruct the commanded position (more on that on the post-processing section 4.3.1).

The Picoscope 5443D comes with 4-channels. For the first experiment with a sine excitation, two channels are used for the sensor channels output (R_1 and R_2 as seen in Fig. 3.1b). The third channel is used for the encoder readout. Last, the fourth channel is used for the reference measurement. The goal of the latter is to allow *AC* measurements and to be further used towards removing any noise of the power supply.

For the remaining measurements, which include a decaying sine, a frequency sweep and a white-noise induced sine the HandyScope HS5 is used. The HS5 comes with 2-channels as a single module, but parallel stacking of multiple modules is allowed, therefore allowing more channels depending on the number of available devices. For this case, two modules are available. Matching then the 4-channels of the PicoScope.

During the second set of measurement experiments, it was noted that the ability to remove noise by the use of the reference channel is limited. Moreover, the experiment have been further constrained into purely *DC* measurements. With the fourth channel free at this state, the encoder measurements used two scope channels, simplifying post-processing of the encoder signal.

More details about the technical specifications of both devices are in [44] and [45].

4.2.2 Linear actuation

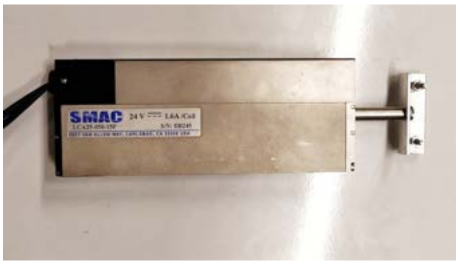


Figure 4.5: SMAC LCA25-050-15F.

Figure 4.6: SMAC single axis controller.

The SMAC LCA25-050-15F (Fig. 4.5) is a 24 V solenoid based linear actuator with a 50 mm stroke. The actuator can be either position or force controlled and has the capability of continues force loading of 6 N or constant force of 9.5 N/A.

The actuator is controlled with a dedicated controller (LC-10)(Fig. 4.6) employing the Embedded Motion Control Library (EMCL) which controls the system in a feedback-loop scheme as described in Fig. 4.7. The controller communicates via a USB serial interface with a PC where several options for operation are included. First option is the provided Graphical Control Interface. Another option is to manually open a communication channel and supply the controller with the appropriate commands as found in the manual [6].

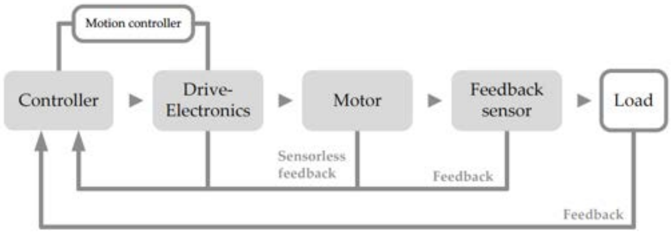


Figure 4.7: SMAC feedback control scheme [6].

For these experiments a combined approach is followed. For the actuator initialization the provided interface is used. Then, custom scripts (provided), with two different implementations (position & force control) are used.

During all the experiments the SMAC is calibrated with a set voltage at 25.2 V.

4.2.3 Software

Connecting all the parts on the software side is done through the `MATLAB` environment. The provided SMAC control scripts as well as the custom dedicated drivers for both acquisition devices are written in `MATLAB`. Moreover, offline post-processing of the data as well as the development of a non-linear model (described in section 5) are all written in the same environment.

Crucial snippets of the code developed during this assignment are included in the appendix section 9.6.

4.2.4 Layout

The assembly of the complete set up is based on a 50 cm × 50 cm × 3 cm aluminum plate which incorporates 5 mm threaded bores placed in a symmetric grid along the surface. This part allows for quick & easy assembly of different geometrical structures based on single blocks of different size. Therefore, in a modular manner, a variety of different loading experiments can be achieved.

For the current experiment, the goal is to apply a mechanical load in a single direction, along the Z axis as shown in Fig. 3.1a. Rectangular blocks are used to fix the parts in place in such a way that the SMAC applies a force along the sensor Z axis. The sensor is then clamped in a dual-purpose custom vice. The vice acts both as the mechanical constrain, blocking translation and rotation along the X/Y axes and also incorporates the electrical connection pads. Here, the electrical contact between the sensor conductive traces (carbon black filled TPU) and the measurement system is done with a flexible conductive copper tape which is attached on the clamp's inside surface. The clamping force acts a pre-load to ensure sufficient contact.

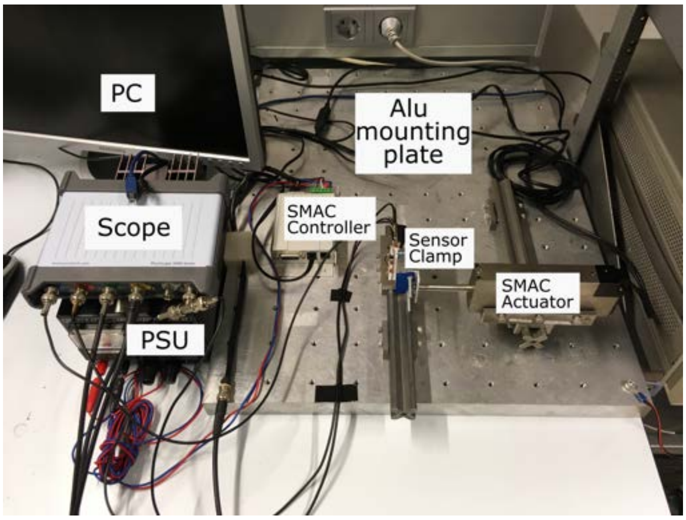


Figure 4.8: Actual experiment setup.

4.3 Measurements

4.3.1 Sine

The 4^{th} sensor sample is used for the first experiment. The SMAC actuator configuration can be found in Appendix 9.7

To read out the sensor, it is placed in a half-bridge configuration using a 10 kΩ resistor and a 2 V DC voltage source. The half-bridge output voltage is measured using the Picoscope 5443B set at a sample frequency of 2 MHz.

Following the discussion under the *Oscilloscope* section, the encoder signal is fed into the third channel as means to obtain proper time synchronization.

The actual sampling interval returned from the driver [42], is $\Delta t_s = 5.0400 \cdot 10^{-7}$. This yields an actual sampling frequency of 1.9841 MHz during streaming. A `hardwareDownSampleRatio` of 20 is defined in the driver configuration [42]. The time-step of the discrete system is then $\Delta t = \Delta t_s * \texttt{hardwareDownSampleRatio} \Rightarrow \Delta t = 1.008 \cdot 10^{-5}$. At the sensor channels, an additional down-sampling ratio of 5 is applied.

Moreover, also digital filtering is applied. The description of this filter will be discussed after the measurements are obtained since it depends on the frequency analysis of the sensor's response.

Due to an observed 78 kHz interference of the SMAC controller, a hardware RC low-pass filter is added to the sensor measurement channels with $R = 10\,\mathrm{k}\Omega$ and $C = 15\,\mathrm{nF}$ with a cut-off frequency $f_c = 1.0585\,\mathrm{kHz}$ effectively leading to a noise suppression of ≈ 37.35 dB. For clarity, a block overview of the complete setup revealing the top-level connections is presented in Fig. 4.9.

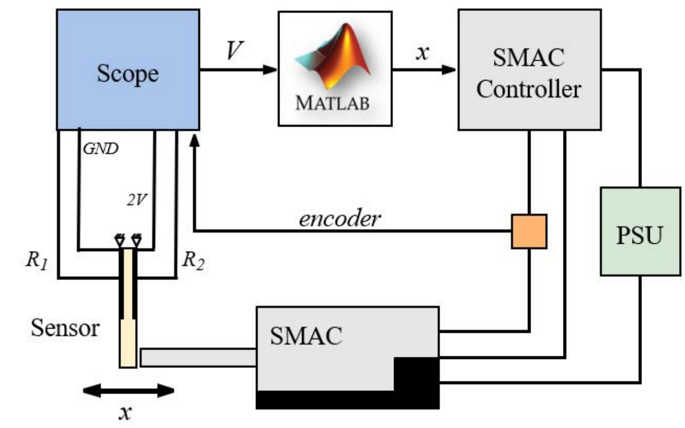


Figure 4.9: Overview of the experiment set-up (orange box represents the *splitter* cable for the extraction of encoder's channels A & B signals).

With this setup, two experiments have been conducted, each of 5 min duration. One measurement set is to be used for model training and the second for validation (more in chapter 5). From each experiment three data sets are to be obtained. From the perspective of the PL model, the first set $\hat{e}(t)$ represents the SMAC position and $\hat{i}_1(t)$ and $\hat{i}_2(t)$ which represent sensor response (half bridge output voltage for each trace).

Results

In order to avoid conclusions that will not hold , there is one crucial point that needs to be clarified. Due to the measured quantities being displacement (input) and voltage (output) a domain coupling is present between the mechanical and electrical domain. Therefore, any non-linearity or any insight revealed by processing these data, is not appropriate for characterizing either the mechanical nor the electrical behavior of such sensor.

Therefore, the conclusions drawn within this assignment have the goal to describe the 3-D printed sensor behavior from a phenomenological view and not to analytically describe the physical behavior of the sensor. For the latter, the first step would be to work in the domain of interest, e.g for the mechanical hysteresis, either by deriving the relation between the measured voltage and the force applied to the material, or by contacting force-controlled experiments with the displacement being the measured quantity or vice versa.

The last step before obtaining usable results is to apply a digital filter to clean out the noisy data. Since all the post-processing takes place offline a zero-phase filter is employed by filtering once in the forward and then in the reverse direction. The filter chosen is a 2$^{\text{nd}}$ order Butterworth filter. In order to select the appropriate cut-off frequency the frequency analysis of each signal is considered, as seen in Fig. 4.10

From Fig. 4.10 it can be assumed that a minimum cut-off frequency of $f_{c_{\min}} \gtrapprox 8$ Hz is needed so that the higher harmonics are included up to the point that the magnitude of the highest harmonic is buried under the signal noise. Then, by conservatively choosing a cut-off frequency of f_c =20 Hz a reasonable balance between the noise and sensor bandwidth is obtained. The filter can then be expressed with a transfer function (in the z-domain) as:

$$B_2(z) = \frac{0.0626 \cdot 10^{-5} + 0.1252 \cdot 10^{-5} z^{-1} + 0.0626 \cdot 10^{-5} z^{-2}}{1 - 1.9978 z^{-1} + 0.9978 z^{-2}} \tag{4.1}$$

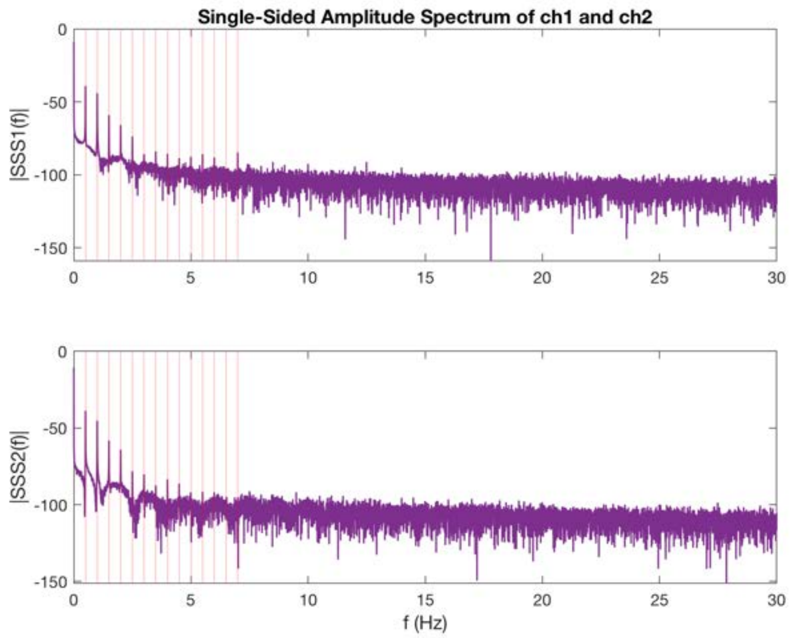


Figure 4.10: FFT of both strain gauges.

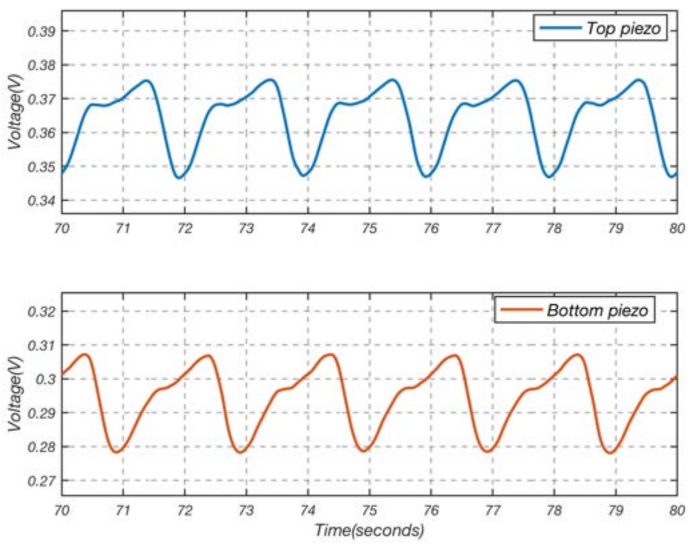


Figure 4.11: Strain sensor response of top and bottom traces.

To obtain the position (input), the measured output signal from the encoder merger is post-processed such that the signal is reconstructed back into the original encoder channels $A+$ and $B+$. To achieve this, a split-logic algorithm is applied to the measured data which looks ahead in the signal to determine how to assign the appropriate normalized state 0 or 1 to each reconstructed channel, based on the four different voltage states (Fig. 4.12(a)). With the reconstructed channels available (Fig. 4.12 (b)), the original excitation signal (Fig. 4.12 (c)) is obtained through a Finite State Machine (FSM) given an increment of 5 nm for each encoder pulse (one for each channel state change (quadrature)). The results are visualized in the following figures.

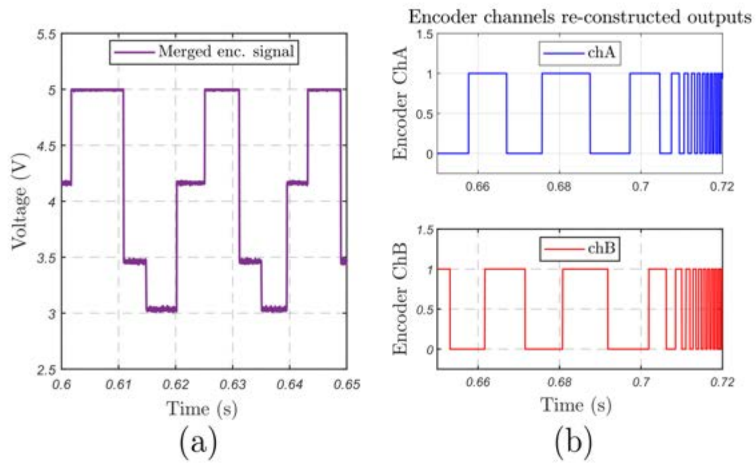


Figure 4.12: (a) Merger output, (b) Re-constructed Encoder channels, (c) FSM output position.

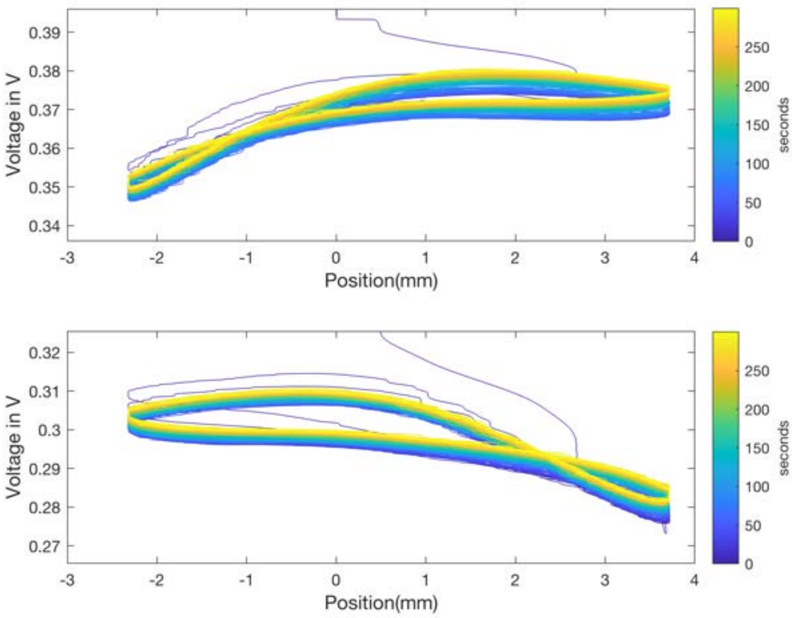


Figure 4.13: Strain sensor individual trace input-output relation, R_1 (top), R_2 (bottom).

With the reconstructed position, the input-output relation. when plotted against time (color gradient), clearly shows that the material exhibits a hysteretic behavior. Moreover, it helps to visualize the sensor drift.

The differential measurement $y = R_1 - R_2$ then yields the results of Fig. 4.14, where the non-linearity can be observed with a close inspection of the differences between the commanded position signal and the output measurement, where the sine peaks seem to be in phase but the rest of the signal is not. Here, the second y axis portraits the SMAC position control signal, which is scaled in magnitude for a better comparison.

Last, the input-output plot of the differential measurement reveals the hysteresis loop (Fig. 4.15).

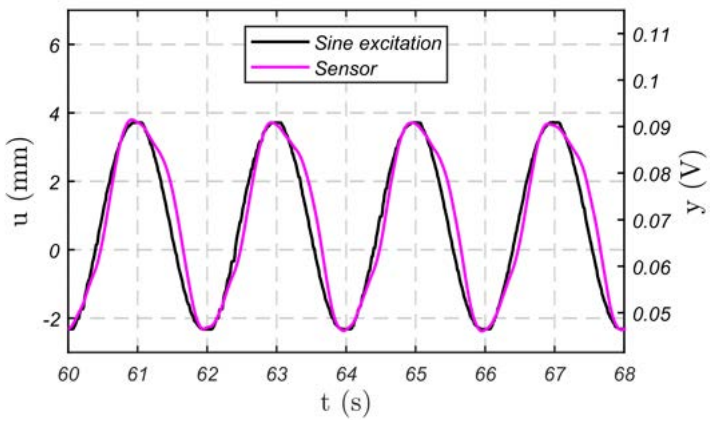


Figure 4.14: Reference (black line) versus sensor output (pink line).

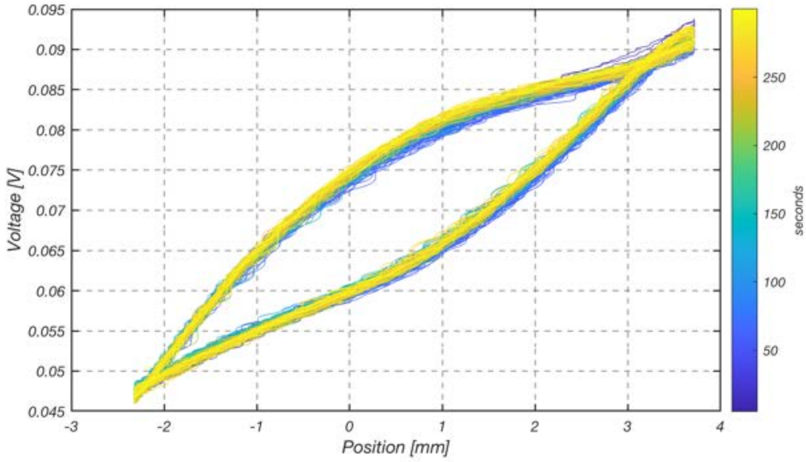


Figure 4.15: Strain gauge differential response to commanded position

The effect of the differential measurement upon the drift can be better visualized by plotting a limited number of cycles with a quite large time interval between the cycles as shown in Fig. 4.16.

To calculate the relative resistance change $\Delta R/R$ from the half-bridge voltage measurements of the strain gauges the resistance is first computed as follows.

From circuit (see Appendix 9.8)

$$V_m = \frac{R_{\text{br}}}{R_{\text{br}} + R_{\text{S}}} V_0 \tag{4.2}$$

Where R_{br} represents the bridge resistance and R_{S} the resistance of the sensor. Solving for R_{S}

$$R_{\text{S}} = R_{\text{br}} \cdot \frac{V_0 - V_{\text{m}}}{V_{\text{m}}} \tag{4.3}$$

With mean $\hat{R}_{\text{S}}$

$$\hat{R}_{\text{S}} = \frac{1}{n} \sum_{k=1}^{n} R_{\text{S}_k} \tag{4.4}$$

With n the number of data points. Finally,

$$\frac{\Delta R}{R} = \frac{R_S - \hat{R}_S}{\hat{R}_S} \tag{4.5}$$

Where R_{br} is the $10\,\mathrm{k\Omega}$ used in the half-bridges, $V_0 = 2\,\mathrm{V}$ the DC supply voltage and V_m stands for the sensor measured voltage data (for R_1, R_2 and $R_1 - R_2$). From 4.14 the period of $T = 2\,\mathrm{sec}$ can be extracted. The relative resistance change $\Delta R/R$ of each gauge can then be plotted for specific load-unload cycles (instead of the entire measurement) as means to increase the clarity of the results (Fig. 4.16).

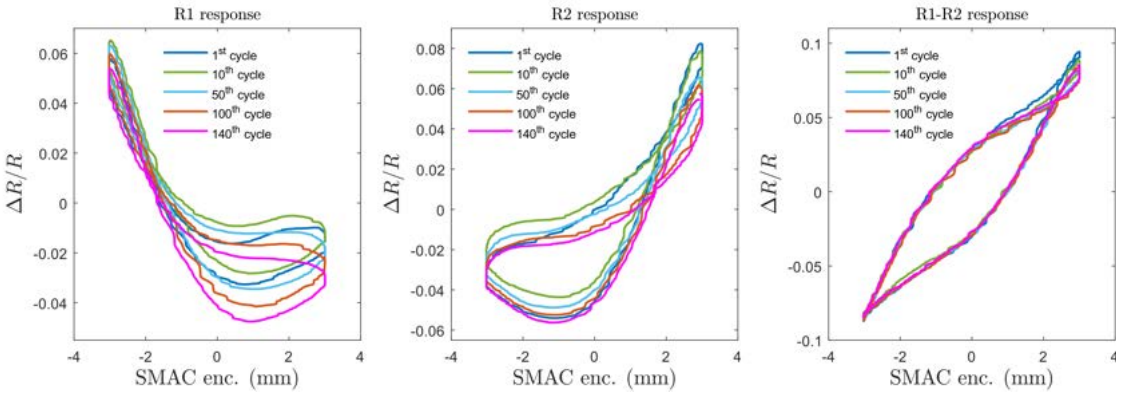


Figure 4.16: Relative resistance change over SMAC position for 5 cycles.

In the current scenario, in contrast to the results of [14], the differential measurement alone is not enough in order to linearize the sensor output. Of course, the differential measurement when compared to Fig. 4.13 provides a more usable signal where the loop crossing is less emphasized and it additionally improves the sensor readout in terms of measurement noise as well as, in certain cases (as in the current one), it cancels out a significant amount of drift (Fig. 4.16). Although, the latter is found to not always be true, therefore not fully reproducible. Further research is required in order to properly model the drift behavior of such sensors.

4.3.2 Various excitations

Moving on, the following experiments aim to produce the required data for a thorough validation of the model that is to be developed. With the goal to develop a model that can capture the non-linear behavior of the sensor, it is important to obtain data that will reveal the non-linear response of the sensor under various circumstances, typically more complex than a simple sine wave.

From the literature, one of the main attributes of the hysteresis models, is the ability to capture the behavior of the minor-loops. Therefore, experiments with an excitation signal of a decaying sine wave can be considered sufficient for that purpose.

Due to the stated non-linearities, convenient methods such as an impulse response to validate the systems response under any excitation are not possible. Although, the sensor's response can be validated for a given range of frequencies via a frequency sweep experiment. The data obtained from the latter can also be considered as a benchmark to obtain the functional frequency range. Moreover, with a given frequency band in mind, a true random signal, such as a pure Gaussian noise generated excitation can potentially cover any expected system response within the used frequency band.

To sum up, all the required data for validating a model in the context of this assignment can be obtained with three distinct experiments given the following excitations:

- Decaying sine-wave
- Frequency sweep (or chirp)
- Gaussian noise

The gathering of these data, along with the model performance which will be discussed later, can be found at [9]. The exact excitations used for each signal can be summarized as follows [9]:

1. A decaying sine wave with a frequency of 0.5 Hz and an amplitude that exponentially decreases over 200 s from 2 mm to 100 µm.
2. A frequency sweep with a frequency exponentially increasing over a period of 100 s from 0.1 Hz to 3 Hz and an amplitude of 3 mm.
3. Gaussian noise filtered with a 500$^{\text{th}}$ order FIR low-pass filter with a −3 dB cut-off frequency of 4 Hz

Similarly to the first sinusoidal excitation experiment, a 2$^{\text{nd}}$ order 20 Hz zero-phase Butterworth filter is applied. During this second round of measurements, the Handyscope HS5 (Fig. 4.2) is used. The sampling frequency is set at 6.25 MHz. The half-bridge readout resistors are switched to lower value ones with R=560 Ω, due to the aging effects as described in section 3.3.1, as means to match the sensor trace resistances, and the DC supply voltage is set at V_s =1 V. Additionally, all three following signals are downsampled to 40 Hz [9]. Last, the computation of $\Delta R/R$ follows from equation 4.5.

Following are the results of these measurements and as expected, for the decaying signal, the forming of minor loops is present. The frequency sweep is limited to maximum frequency of 3Hz and subsequently terminated at 150 sec since the higher excitation frequencies did not provide usable results. This may be due to the unstable behavior of the SMAC actuator towards higher frequencies, where the tuned velocity and acceleration profiles where not calibrated properly.

Decaying signal

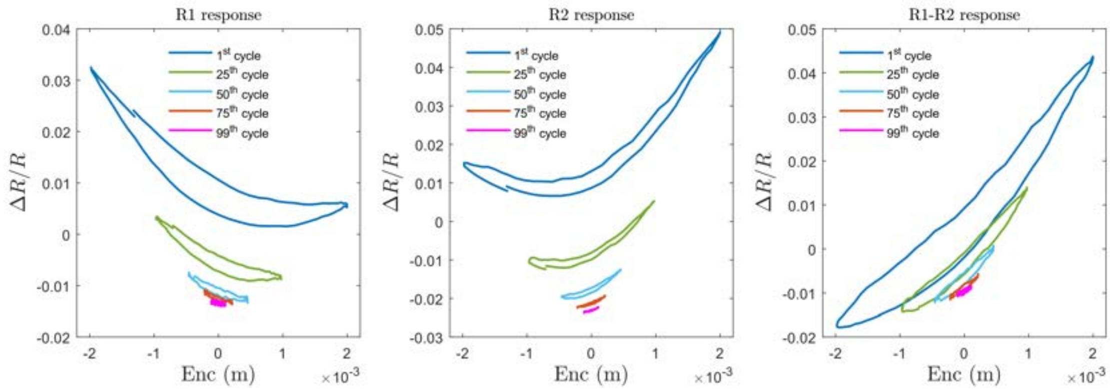


Figure 4.17: $\Delta R/R$ versus commanded position for 5 equally spread cycle [1].

For the decaying sine-wave, looking at Fig. 4.17 the significant drift in the sensor response is clear and in this case, the differential measurement, although it improves the response in that regard, it is not able to completely eliminate the drift such that all the minor loops lie inside the major loop.

[1] 1$^{\text{st}}$ cycle represents the first loop of the processed data, meaning that this loop occurs at t =50 sec of the experiment.

Forming of minor loops that are subject to such drift in the response, renders the existing models short-handed since all of the models described in the literature that are able to capture minor-loop behavior, focus on the branching that occurs inside the confined space defined by the major loop. Meaning, that all models covered by the overview of the chapter 2 are prone to not accurately modeling such responses.

The following plots for the measurements of both the sweep and noisy excitation include the entire set of gathered data due to the varying frequencies.

Frequency sweep sine

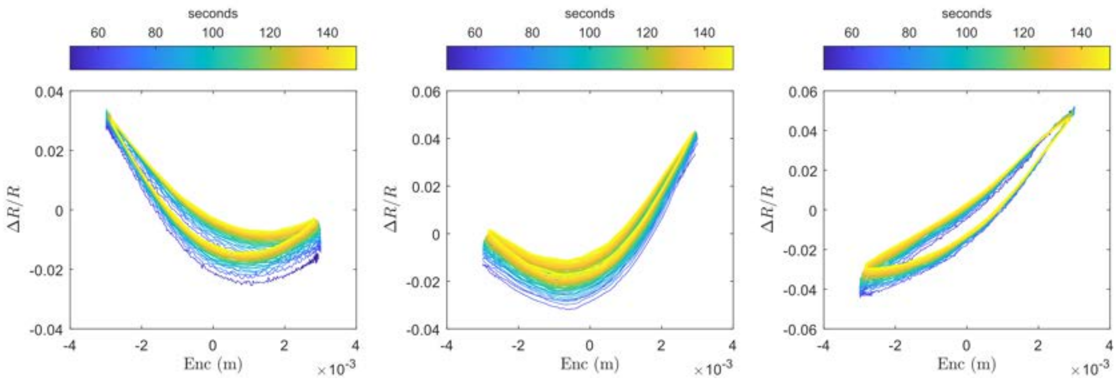


Figure 4.18: Frequency sweep $\Delta R/R$ versus commanded position. (a) R_1, (b) R_2 and (c) $R_1 - R_2$.

Gaussian noise

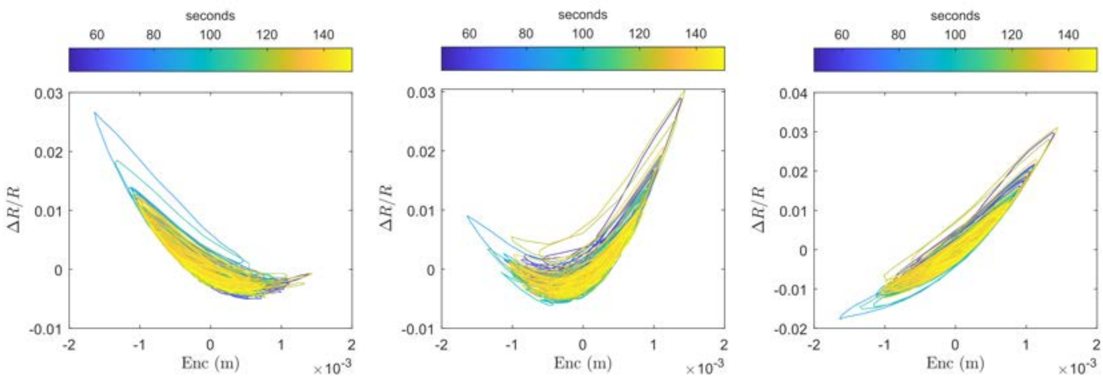


Figure 4.19: Noise excitation $\Delta R/R$ versus commanded position. (a) R_1, (b) R_2 and (c) $R_1 - R_2$.

A remark that can be obtained following the observation of the measurement results of all excitation signals, is that in the differential sensor, the strain-gauges reveal an opposing behavior, as expected, where the hysteretic non-linearity seems to be more emphasized towards the contraction maximum ($+3mm$ for R_1 and $-3mm$ for R_2) of each gauge and the response shows a less hysteresis towards the expansion maximum ($-3mm$ for R_1 and $+3mm$ for R_2), as illustrated in Fig. 4.20. Although, towards the expansion maximum, a loop crossing is present which is persistent after the differential measurement as shown in Fig. 4.16. This crossing in the response is often observed within the NIFTy group experiments [14, 46]. Despite the multiple observations, the reason behind it is unidentified. However, following the results of [9] (see Figs. 4.17-4.19) it does not appear to be present in all measurements.

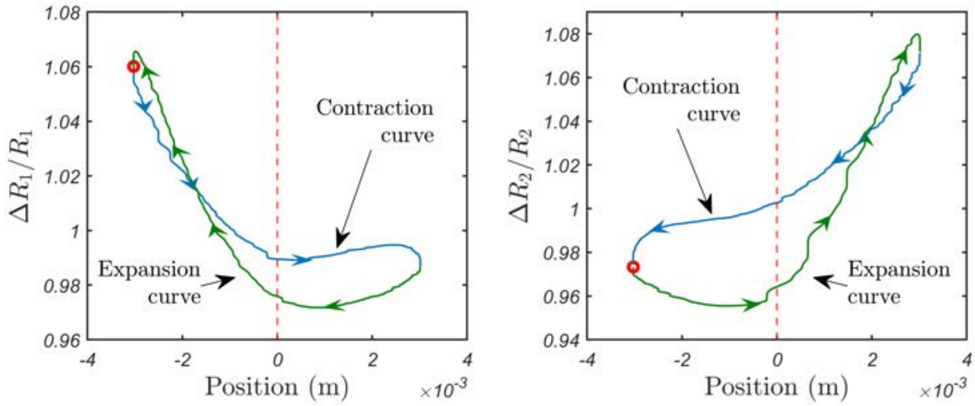


Figure 4.20: Illustration of contraction (blue) and expansion (green) phases of each strain-gauge. Data: isolated 10^{th} loop response of R_1 and R_2 from the sine experiment (direction arrows with help of [7]). Dotted red line represents the resting position of the sensor and red cycle indicates the first data point of the loop.

Following the conclusions of [14] the importance of sensor symmetry is clear: the more symmetrical the gauges are, the better the *cancellation* of each other non-linear response is, hence for future development of such sensors more focus should be on sensor symmetry. Two suggestions can be made towards further improving the sensor symmetry: (a) fine-tuning the PVA raft-Ironing combination measure (sec. 3.4) by adjusting the offset between the ironed layer and 1^{st} sensor layer, or (b) by printing the sensing element within the sensor body, meaning that (at least) the first printed layer of the sensor will be a non-conductive material. With the latter, the extra challenge of attaching the measurement cables is added.

4.4 Conclusion

An experimental setup to characterise the 3D printed tactile sensor is described and an overview of obtained measurement results is presented. The excitation configuration is discussed and digitally implemented. Data are gathered for the sine-wave excitation and a low-pass filter is described. The response from each trace is logged from which the differential output is computed and the hysteresis is revealed. The work towards a decaying sine, a frequency sweep and a noise induced signal is briefly described further validating the hysteretic response as well as more clearly showing the sensor remaining drift that persists in the differential measurements. The processed data are ready to be used for training and validating the model derived in the following chapter.

5 Hysteresis Modeling and Compensation

5.1 Introduction

With the measurements obtained, the next task is to develop a model that can capture or mimic the non-linear behavior of the sensor to a satisfactory degree. To avoid over-complexity, the focus is shifted towards the hysteretic non-linearity, and rate-depended phenomena such as creep or stress relaxation, as well as sensor drift are neglected. However, even though these phenomena are not directly modelled, they are still part of the measurement data that was gathered (more on this in the model selection section). Following from the domain coupling as explained in the results section of 4.2.1, there is not a direct way to distinguish these effects with the current experimental setup. Therefore, it is important to point out that even though the focus is on the hysteresis, additional non-linearities are inherent in the data, but they cannot be distinguished, hence separately model each one. Albeit, given that these non-linearities are present in both strain-gauges the more identical (symmetry) they are, the more these effects are cancelled when a differential measurement is considered. The task is then categorized as follows:

1. Select a suitable model from table 2.1 to build upon.
2. Insert measurement data in the existing black-box implementation of the model[1] as proof of concept.
3. Model development and modification.
   (a) Modified model analysis.
   (b) Parameter estimation.
   (c) Model inverse (Compensation).
   (d) Model stability analysis.
4. Discuss on results and model performance.

5.2 Model selection

Following the insight from [18] it is known that due to the materials involved in the sensor, rate-depended phenomena such as stress relaxation and creep are inherent in the system. Aiming for the model with the most potential on accurate representation of the sensor, even though the focus is not on analytically addressing the rate-dependency it is beneficial to build upon a model that incorporates components able to capture this behavior.

This insight effectively limits the available models to the ones of Chua & Stromsmoe, Kuhnen and the Power Law model. The Chua & Stromsmoe model is limited to capturing hysteresis with local memory only. Even though this is computationally less expensive compared to models based on a stronger notion of global memory, with the nowadays available computation power in commonly used PC's and considering the fact that during the development phase, the model fitting procedure is to be handled offline the Chua & Stromsmoe model is ruled out as well.

The final selection between the Kuhnen and the Power-law model is more challenging. Both models carry the same feature set although they are developed in different ways. The roots of both models lie in the Preisach model, although the Kuhnen model has a stronger relation to the original implementation through the work of Krasnosel'skii-Pokrovskii as well as the work of Prandtl-Ishlinskii where the approach is based on an operator-based concept, as explained

[1] With the help of [47].

in the literature section. The Power-Law model on the other hand, follows a circuit model alternative representation. The decision is made to work with the Power-law model for the following two reasons. First, being a circuit model, it is more straightforward to analyse it through the circuit dynamics and second, following the comparison of [5] the Power-law model is less-sensitive to parameter changes and similarly accurate to the one of Kuhnen with considerable lower number of parameters.

5.2.1 Black-box Power-Law

A first hint as to whether the model selection is valid, can be obtained by plugging the obtained measurement data into the black-box Power-law model provided by [5]. Two simulations are then performed, one for testing the model with a single cell, therefore with minimum number of parameters, and a second one where the number of cells N is increased until no substantial performance boosts are obtained.

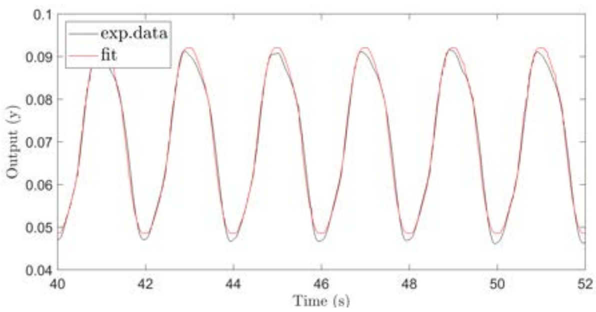


Figure 5.1: Hystool 1 cell simulation response.

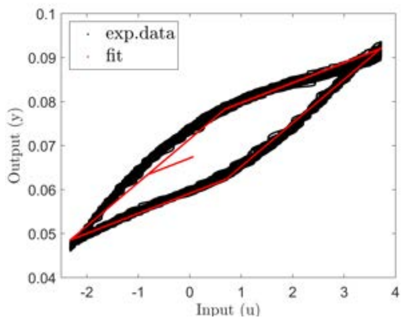


Figure 5.2: Hystool 1 cell fitted loop

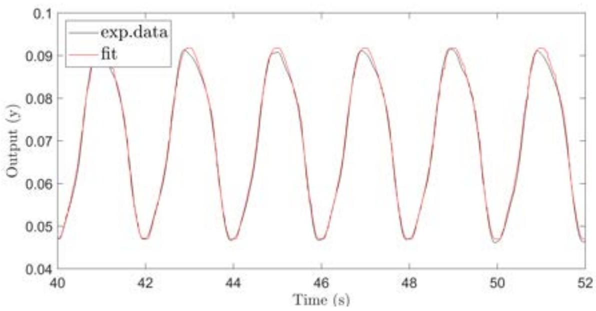


Figure 5.3: Hystool 10 cell simulation response.

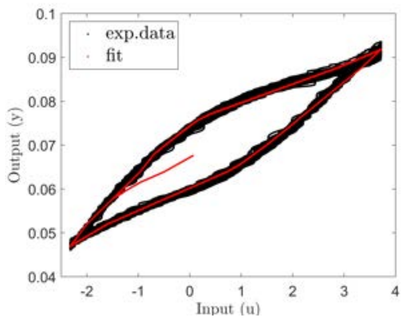


Figure 5.4: Hystool 10 cell fitted loop

It is immediate that out of the box the Power-Law seems to be able to capture the hysteretic behavior of the sensor. A single cell model with 6 parameters for the estimation (w_0, w_1 of the VCVSs, the I_S, P of the non-linear resistor d_N, the voltage bias V_b and last, the α_j coefficients of the non-linear resistor of the second sub-circuit), is able to correctly scale to the data (Fig. 5.2), although the loop's corner points are slightly misaligned and half of the unloading path of the fitted model is in an offset to the original data. A second simulation is then run with a 10 cell Power-law model. The resulting parameters are now 19 and the fitting is significantly improved as seen in Fig. 5.4.

Table 5.1: Hystool Simulations overview

					Toolbox Parameters		
Model	Cells (N)	No. pars	e_{rms} %	e_{max} %	M	S	p
Power-Law	1	6	2.799	15.555	1	0.001	20
Power-Law	10	19	1.746	15.208	5	0.001	20

The *Toolbox Parameters* are defined in [47]. Briefly, M represents the number of PWA basis-functions that describe the model output (current of non-linear resistor of sub-circuit (b), see Eqn. 2.17. S and p are toolbox representation for the estimation parameters of the non-linear functions that are optimized either through a *brute-force* or the *fmincon* algorithm [15].

The above results, provide enough confidence that the Power-law model is a good candidate for capturing the non-linear response of 3D printed sensors. Therefore, the next task is to develop the model from scratch in order to more thoroughly analyse it and incorporate it in a given application.

5.3 Modified Power-Law

5.3.1 Development

An overview of the original implementation can be found under the literature section 2.1.7. In the context of this assignment the decision was made to start as simple as possible and further build on complexity depending on the performance. Again, starting with the simple sine measurements, and then expanding towards more complicated excitations.

The first simplification step can be seen in Fig. 5.5 where a form of the circuit whose dynamics are more straightforward, is described. The first striking difference is the decision to discard the non-linear resistor of the sub-circuit (b) and represent it as a linear one. The insight for this move stems from the fact that the sub-circuit (a) network comprises the main building block for modeling hysteretic complexities as well as the rate-depended behavior when an odd power function is used as part of the representation [15], where the sub-circuit (b) main functionality is for the identification procedure which will also change in contrast to the original implementation.

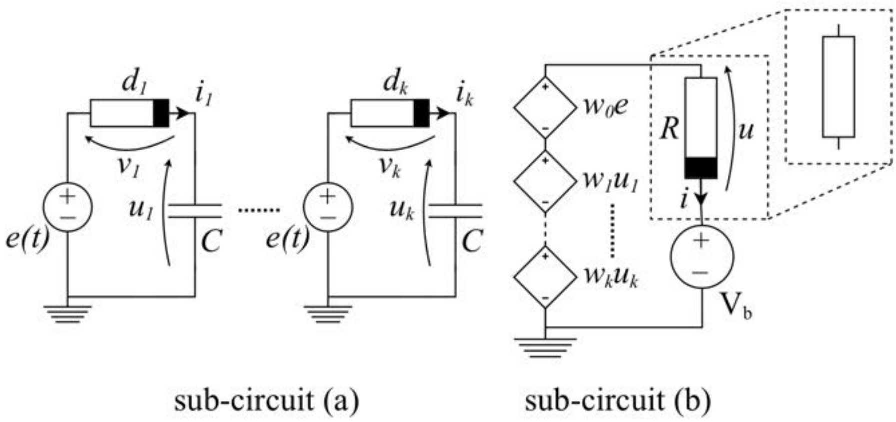


Figure 5.5: Modified Power-law model.

Second, following Fig. 2 of [15] where the characteristic of the non-linear resistor of the sub-circuit (a) is represented with a strictly increasing piece-wise function, the choice is made to represent the resistor with a continues strictly increasing function across the whole range. Aiming for shape similarity, a good candidate is then the hyperbolic sine function.

$$i_k = f(v_k) = I_{S_k}\left(e^{\frac{v_k}{V_{T_k}}} - e^{\frac{-v_k}{V_{T_k}}}\right) \tag{5.1}$$

Where I_{S_k} and V_{T_k} are the resistor shaping parameters (unique for each cell), e is Euler's number and v_k is the voltage across the non-linear resistor.

Of course, the selected form is a choice, and as shown later, assuming polynomial representations will yield similar and in some cases better results.

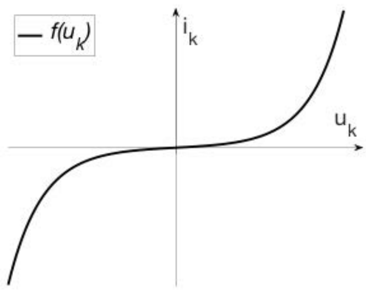


Figure 5.6: Hyperbolic sine form of the non-linear resistor d_k.

Moving on, with the modifications in place the circuit dynamics are addressed.

First the model output should be defined. Repercussions of the first simplification is that by choosing a value of the linear resistor as $R = 1$, the model output can be chosen to be either the current, as the original implementation, or the voltage across the resistor. Therefore, assuming voltage as the model's output, from kirchhoff's voltage law (KVL) for sub-circuit (b) it follows:

$$iR = w_0 e + \sum_{k=1}^{N} w_k u_k - V_b \tag{5.2}$$

Where, V_b is the voltage bias of sub-circuit (b). Similar to the original PL, the weights $w_0 ... w_k,\ k = [1...N]$ as well as V_b are to be estimated during the identification.

The voltage u_k across the capacitor for a cell can be calculated from the natural response of the system. Starting from Kirchhoff's current law (KCL) for the sub-circuit (a) the following first order differential equation is obtained:

$$C_k \frac{du_k}{dt} = i_k \tag{5.3}$$

Substituting with $f(v_k)$ from 5.1,

$$\begin{aligned} C\dot{u}_k &= f_k(v_k) \\ \dot{u}_k &= \frac{1}{C} f_k(v_k) \end{aligned} \tag{5.4}$$

Then from KVL for sub-circuit (a):

$$\dot{u_k} = \frac{1}{C} f(e - u_k) \tag{5.5}$$

From the definitions of 5.2 and 5.5 the state-space representation of the modified model can be formed, with $\dot{u_k}$ the state variables.

Towards the digital implementation, for each iteration the state is computed via the Forward Euler Integration method:

$$u_{k_{n+1}} = u_{k_n} + \frac{1}{C} f(e_n - u_{k_n}) dt \tag{5.6}$$

With n the sample number.

Knowing that voltages u_k are to be weighted by the estimated weights through optimization the assumption is made to fix all $C = 1$. Although this simplification is trivial since the time-constant of each cell is depended on the product $d_k C$ which eventually is shaped during parameter estimation. In the diagram of Fig. 5.7 the flow of the digital implementation of the forward mPL model focusing on post-processing is shown.

5.3.2 Identification

With the model equations ready next step is the estimation of the remaining weights. Similarly to the original PL, sub-circuit (b) includes the main parameters for the identification. This means that the main parameters to be estimated are the weights of the VCVS's and the voltage bias V_b. Additionally, the parameters V_{T_k} and I_{S_k} of the non-linear resistors of each cell are chosen to also be included in the optimization. This effectively increases the amount of parameters per cell but simultaneously reduces the number of cells necessary for good fit (as shown later in Sec. 5.3.3).

For the identification MATLAB's function `fmincon` from the Optimization Toolbox is employed, which is defined as a non-linear solver that finds the minimum of a constrained non-linear multi-variable function [35]. In this case it is configured to work with an interior-point algorithm. Briefly, an Interior-point algorithm is given the task to minimize a function f such that

$$\min_x f(x), \quad \text{subject to } h(x) = 0,\ g(x) \leq 0 \tag{5.7}$$

where $h(x)$ and $g(x)$ are the constraints, which in this case are left to their default values (internally computed). The problem can also be represented in the following form [35]

$$\min_{x,s} f_\mu(x,s) = \min_{x,s} f(x) - \mu \sum_i \ln(s_i), \quad \text{subject to } h(x) = 0,\ g(x) + s = 0 \tag{5.8}$$

The approximate problem of 5.8 represents a sequence of equality constraints instead of the inequality of 5.7 effectively rendering the solving easier [35]. In the latter equation, the logarithmic summation term is the *Barrier function*, the s_i are the *slack variables* and they are restricted to be positive as means to keep the barrier function positive. Moreover, the number of the slack variables is defined by the number of the inequality constraints of g. For a more in

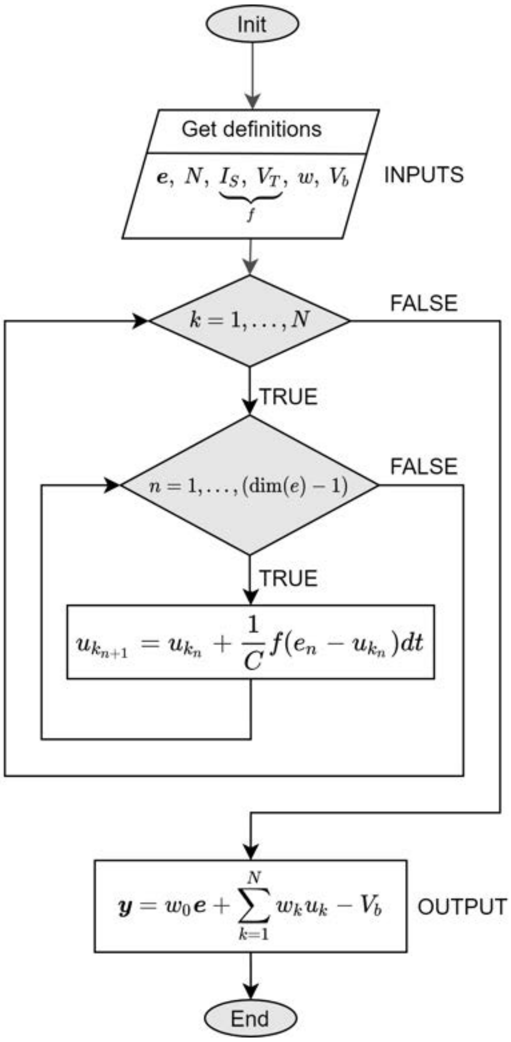


Figure 5.7: Process flow of the digitized mPL algorithm. N denotes the number of cells and $\dim(e)$ denotes the length of the input vector. Code in 9.6

depth description of the inner workings of the solver, the reader is advised to read through [35] and references therein.

Back to the modified model, as stated, the choice can be made to include in the optimization the parameters of the non-linear resistors of the first sub-circuit. Meaning that the parameters I_{S_k} and V_{T_k} will be passed as estimation variables in the multi-variable problem, summing up to a total of $3N+2$ total parameters. Subsequently, optimizing the non-linear resistor parameters the shape of its characteristic curve it is expected to change as well. The effect of this shaping through optimization, can theoretically lead to a fewer number of required cells for a given performance in comparison to the original PL.

For this minimization problem, a good cost function candidate that is often used is the sum-squared error (SSE), in this case defined as

$$SSE = \sum_{t=0}^{t_{\text{end}}} \left(\hat{i}(t) - i(\hat{e}(t))\right)^2 \tag{5.9}$$

where $\hat{i}(t)$ is the measured output, and $i(\hat{e}(t))$ the model output for the input $\hat{e}(t)$.

5.3.3 Fitting

The last step is to insert the measured input data in to the model, and plot the modelled against the measured output. First of all the two obtained datasets (input-output) for the sine excitation are then used. Both datasets, one used for fitting and one for the compensation/validation to come, are normalized as follows:

$$\begin{aligned} e_n &= \frac{e_n}{\max|e|} \quad \text{with} \quad n = [1 \cdots \dim(e)] \\ i_m &= \frac{i_m}{\max|i|} \quad \text{with} \quad m = [1 \cdots \dim(i)] \end{aligned} \tag{5.10}$$

The initial conditions can be found under table 5.2.

Then using the initial weights the optimization is subjected to a single constraint $b = 10^2$ that defines the maximum of the sum of the optimized weights.

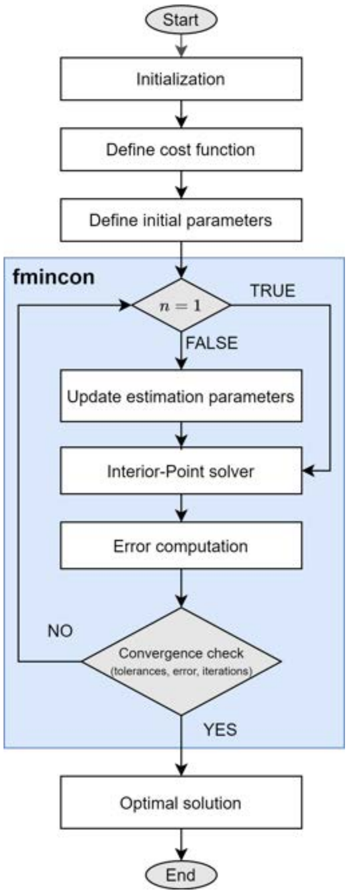


Figure 5.8: High level flow chart of the fmincon based parameter identification.

Finally, the input data is applied to a single cell ($N = 1$) modified PL (mPL) model given the optimized weights and parameters.

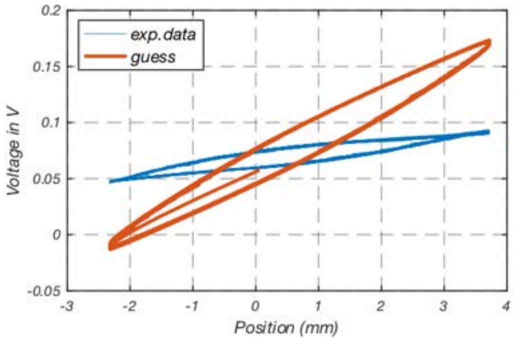


Figure 5.9: Model output for initial conditions.

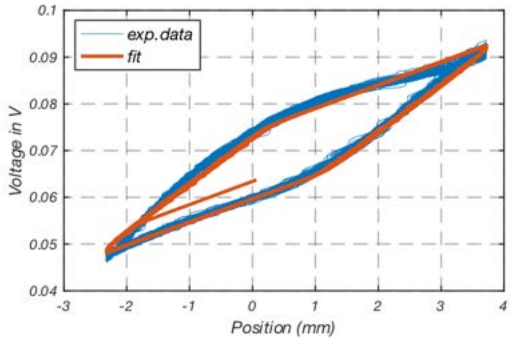


Figure 5.10: Model output with fitted weights.

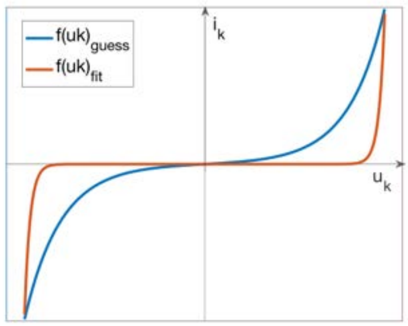


Figure 5.11: Non-linear resistor shape after optimization.

The modified PL seems to be able to capture the non-linear response of the sensor even after the proposed simplifications. Table 5.2 describes the weights pre and post optimization and table 5.3 summarizes the single cell mPL performance.

Table 5.2: Estimation Parameters

Parameter	Initial condition	Fitted weights
I_S	0.1	1.72e-07
V_T	0.2	0.0307
w_0	[1, 0.4]	[0.1972, 0.1975]
V_b	-0.6	-0.6775

Table 5.3: Single cell PL vs mPL performance.

Model	Cells	No. pars	e_{rms} %	e_{max} %
PL	1	6	2.279	15.555
mPL	1	5	1.369	11.867

The errors described in the above tables are defined as follows for a direct comparison to the black-box model with the errors provided by the Hystool Toolbox.

$$\begin{aligned} e_{\text{rms}}(\psi,\hat{\psi}) &= 100\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\psi(i)-\psi\hat{(i)})^2} \\ e_{\text{max}}(\psi,\hat{\psi}) &= 100(\max|\psi(i)-\psi\hat{(i)}|) \end{aligned} \tag{5.11}$$

Moving on, for fitting performance evaluations the coefficient of determination R^2 is chosen, which, in contrast to the RMSE that indicates the absolute[2] fit of the model, it is a relative indicator of the goodness of the fit in a scale of $[0, 1]$ with 1 indicating that the model can predict all the variability of the response data.

From table 5.3 it can be observed that the mPL model performs better than the original PL model with the same number of cells and for one parameter less for estimation. This result may stem partly from the optimization of the non-linear resistor and the continuous hyperbolic sine function representation. Although, this effect may be emphasized towards higher number of cells since the PL model optimizes the V_T and I_S non-linear resistor threshold through an exhaustive search once and then the same thresholds are applied to all the cells with the voltage threshold kV_T linearly increasing with the cell number $N = (1...k)$, whereas in the mPL the V_{T_k} and I_{S_k} shaping parameters are optimized for each cell individually. Meaning, that for a single cell, the non-linear resistors of both models can be considered optimized. Additionally, the optimization tolerances for the `fmincon` were set considerably high while testing without affecting computation time significantly (a local minimum is always found within similar execution times). Since there is no reference as to the configuration of the exhaustive search used by the original model, it is not possible to have a one-to-one comparison of the model itself.

Although the results seem promising, and the mPL model developed is capable of capturing the strain-gauge response, there are a few concerns towards the decision to represent the non-linear resistor as a hyperbolic sine. More specifically, since the function $f(v_k)$ is unbounded, it can be deducted from Fig. 5.11 by looking at first and third quadrant that towards the $\max|u_k|$ values, the current $|i_k|$ starts growing/decreasing in much higher rate than the voltage u_k leading to a steep resistance decrease/increase. While the resistance decreases/increases unbounded so does the time-constant of the circuit, leading to an over-reacting or very slow system that eventually becomes unstable.

To address this issue, in section 5 a stability analysis is described which provides a constraint that avoids the steep resistance decrease, effectively extending the values where the non-linear function does not cause instability. This analysis is also part of the work of [9] where a polynomial representation of non-linear resistor d_k is assumed.

5.3.4 Compensation

The final step is the validation of the model by using its inverse as a means to compensate for the non-linear response. Due to the system being non-linear, common techniques such as an impulse response as means to obtain insight on the systems behavior towards all possible excitations is not adequate. In fact, there is not a single excitation able to provide an insight that can be generalized, it would take an infinite set of measurements corresponding to all excitation to establish the validity of a non-linear model [2]. Therefore, the non-linear compensator's performance shall be validated given certain signals as means to obtain an understanding towards each individual excitation. In spite the fact that a generalized conclusion for the compensator can not be derived, the behavior of the system given common excitations can be tested which should provide a degree of confidence for the use-cases when these excitations are used. This step is done differently for the first attempt towards the sine excitation than towards the various other excitations.

First, the initial work focusing on the sine excitation is discussed.

[2] RMSE can be thought of as a measure of how spread the residuals $((\psi(i) - \hat{\psi(i)}))$ are and how much data congestion is present along the fitting path.

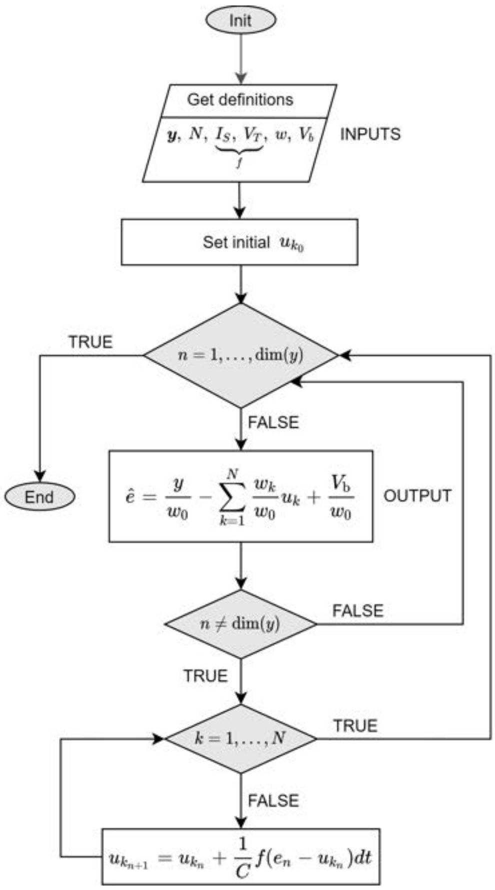


Figure 5.12: Flow chart revealing the process order within the compensator algorithm. Code in 9.6

From equations 5.5 and 5.2 the following state space is formed:

$$
\begin{aligned}
\dot{u}_k &= \frac{1}{C} f_k(e - u_k) \\
iR &= w_0 e + \sum_{k=1}^{N} w_k u_k - V_b
\end{aligned}
\tag{5.12}
$$

Solving for e by simply rewriting this equation such that

$$
\hat{e} = \frac{iR}{w_0} - \sum_{j=1}^{N} \frac{w_j}{w_0} u_j + \frac{V_\mathrm{b}}{w_0} \tag{5.13}
$$

And substitute that to the state

$$
\dot{u}_k = \frac{1}{C} f_k \left(\frac{iR}{w_0} - \sum_{j=1}^{N} \frac{w_j}{w_0} u_j + \frac{V_\mathrm{b}}{w_0} - u_k \right) \tag{5.14}
$$

For the digital implementation of the compensator it is crucial to reverse the order of the compensator, meaning that first the VCVSs of sub-circuit (b) need to be derived and then the estimated input $\hat{e}$ (compensator output) can be computed from KVL. Fig. 5.12 reveals the order inversion of the processes, where y is the filtered differential sensor response. The weights used are the optimized ones from the identification section, as seen in table 5.2.

An important effect of the compensation is the impact on the system's causality. During initialization an array of zeroes needs to be defined with the initial state set to zero. This is okay as long as the system initial state is defined at the origin but extra care is needed when different initial states are present, such as an initial process that synchronizes the sensor and compensator state. This is more crucial if the inverse model is adapted in a real-time computation scheme where the causality issue results in a delay.

With the model set, the second set of data obtained from the measurements is to be used. Here the input data to the compensator is the normalized sensor response and the compensator returns the normalized predicted position $\hat{e}$. The compensator output is then plotted against the measured position e. The results are shown in the following figure.

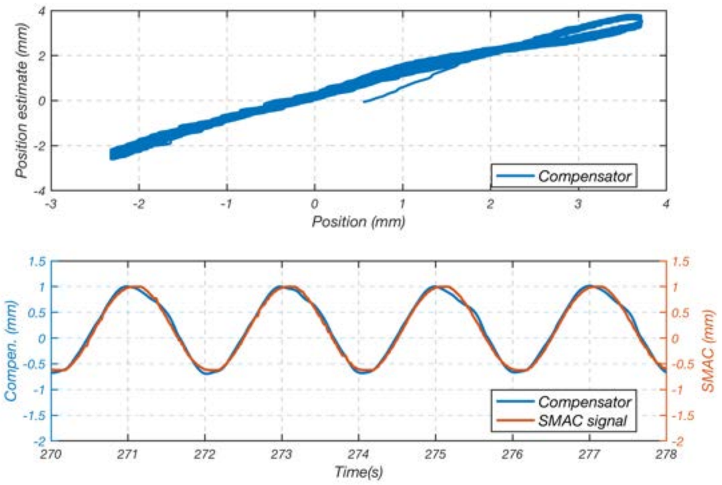


Figure 5.13: Applied compensator prediction vs measured position.

Clearly, it can be seen that the hysteretic behavior is significantly reduced. The predicted position $\hat{e}$ shows a greatly improved match with the SMAC control signal. The plotted results (Fig. 5.13 (top)) have the normalization removed for a better comparison with Fig. 5.10. As a measure of how much the hysteretic response of the sensor is reduced the normalized loop area for all the important signals is calculated using equation 5.15 and the results are summarized in table 5.4.

$$
\begin{aligned}
\hat{A} &= \frac{1}{n}\int_{\hat{x}_{t=0}}^{\hat{x}_{t=n\cdot T}} \hat{y}\cdot d\hat{x} \\
\text{with:} & \\
\hat{x} &= \frac{x-\min(x)}{\max(x)-\min(x)} \\
\hat{y} &= \frac{y-\min(y)}{\max(y)-\min(y)}
\end{aligned}
\tag{5.15}
$$

The strong loop reduction is then further validated, although from the Noise to Distortion ratio it can be seen that the signal non-linearity is not affected. The latter is to be expected as the compensator signal is the inverse of the model that is fitted to predict the behavior of the sensor non-linear response.

In terms of the sine excitation experiments, the only thing left is to incorporate the compensator in a control task and analyze the results and practical effects of the model.

Table 5.4: SINAD & Area.

Signal	SINAD (dB)	$\hat{A}$ (%)
SMAC encoder signal	32.2	0
R_1	4.9	-10
R_2	5.7	15
$R_1 - R_2$	18.4	-18
Fit	23.6	-20
Compensated	20.6	0.8

The discussion towards the more complicated signals follows, as introduced in section 4.2.2.

5.3.5 Equilibrium stability

The work towards stability analysis, fitting and compensation for various excitations is discussed in [9] and briefly described in this report by focusing on the sections of [9] that emerged from co-effort of the authors.

Issue

Before describing the various excitations, the instability issue needs to be addressed. From trial and error simulation towards generating a compensator for signals such as the decaying signal or a frequency sweep, it has been observed that a compensator output is not guaranteed. In these cases the computation starts from the zero state, as the initial step guarantees, although the estimated weights shape the non-linear resistor such that the RC time goes from too low to too fast and eventually, after a few iterations, the compensator state u_k output is out of bounds.

Therefore, it is crucial to obtain a criterion that refrains too steep shaping. In order to improve the fitting performance, in [9] the non-linear resistor of sub-circuit (b) that was initially removed towards the mPL simplification process, is used again. Then, in the new mPL model, both non-linear resistive elements are assumed to be representable with the following polynomial function:

$$f(x) = f_0 + f_1 x + f_2^2 x^2 + f_3^3 x^3 \tag{5.16}$$

By using a non-linear resistor in the second sub-circuit the output of the model is then defined as the current i across the resistor. The non-linear state space then formed is

$$\begin{aligned} \dot{u}_k &= f_k(e - u_k) \\ i &= g\left(w_0 e + \sum_{k=1}^{N} w_k u_k - V_b\right) \end{aligned} \tag{5.17}$$

Direct compensator fitting

Schouten in [9] works around the idea of fitting the compensator model directly to the measured data, instead of predicting the sensor behavior with a model and then inverting it to compensate. This is a more straightforward approach towards a direct application, although it should be noted that the fitted model no longer represents the sensor non-linearity.

Analysis

Before any kind of insight towards the stability of the system can be obtained, it should be linearized around an equilibrium point. Suppose u_{k_0} and e_0 are the state and system input at rest. Then, from the Taylor expansion and by neglecting the higher-order terms, and assuming that in close proximity to the equilibrium u_{k_0} the dynamics of the system are sufficiently described by the first-order terms, the linearized system is described with the following equation:

$$\begin{aligned} \dot{u}_k &= f_{k,0} + f'_{k,0} e - f'_{k,0} u_k \\ i &= g_0 + g'_0 w_0 e + g'_0 \sum_{k=1}^{N} w_k u_k - g'_0 V_b \end{aligned} \tag{5.18}$$

Where:

$$\begin{aligned} f_{k,0} &= f_k(u_{k,0} - e_0) \\ f'_{k,0} &= f'_k(u_{k,0} - e_0) \\ g_0 &= g\left(w_0 e_0 + \sum_{k=1}^{N} w_k u_{k,0} - V_b\right) \\ g'_0 &= g'\left(w_0 e_0 + \sum_{k=1}^{N} w_k u_{k,0} - V_b\right) \end{aligned} \tag{5.19}$$

From the state of 5.18 and for N cells it follows:

$$\dot{\hat{u}} = \begin{bmatrix} \dot{\hat{u}}_1 \\ \dot{\hat{u}}_2 \\ \vdots \\ \dot{\hat{u}}_N \end{bmatrix} = \underbrace{\begin{bmatrix} -f'_{1,0} & 0 & \dots & 0 \\ 0 & -f'_{2,0} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & & -f'_{N,0} \end{bmatrix}}_{\hat{A}} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_N \end{bmatrix} + \underbrace{\begin{bmatrix} f'_{1,0} \\ f'_{2,0} \\ \vdots \\ f'_{N,0} \end{bmatrix}}_{\hat{B}} e + \underbrace{\begin{bmatrix} f_{1,0} \\ f_{2,0} \\ \vdots \\ f_{N,0} \end{bmatrix}}_{o} \tag{5.20}$$

And state space output

$$i = \underbrace{g'_0 \begin{bmatrix} w_1 & w_2 & \dots w_N \end{bmatrix}}_{\hat{C}} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_N \end{bmatrix} + \underbrace{g'_0 w_0}_{D} e - \underbrace{g'_0 V_b + g_0}_{f} \tag{5.21}$$

Looking at 5.20 and 5.21 it can be seen that the linear state is not in the standard $\dot{u} = Au + Be$ form due to the extra o and f terms. The state vector can be augmented to effectively absorb the extra terms into the state and output matrices, such that the extra terms can be incorporated into the state and output matrix:

$$\dot{u} = \begin{bmatrix} u_1 & u_2 & \dots & u_N & 1 \end{bmatrix}^T \tag{5.22}$$

Finally, for the standard form:

$$
\begin{aligned}
\dot{u} &= \underbrace{\begin{bmatrix} -f'_{1,0} & 0 & \dots & 0 & f_{1,0} \\ 0 & -f'_{2,0} & \dots & 0 & f_{2,0} \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & & -f'_{N,0} & f_{N,0} \\ 0 & 0 & & 0 & 0 \end{bmatrix}}_{A} u + \underbrace{\begin{bmatrix} f'_{1,0} \\ f'_{2,0} \\ \vdots \\ f'_{N,0} \\ 0 \end{bmatrix}}_{B} e \\
i &= \underbrace{\begin{bmatrix} g'_0 w_1 & g'_0 w_2 & \dots & g'_0 w_N & -g'_0 V_b + g_0 \end{bmatrix}}_{C} u + \underbrace{g'_0 w_0}_{D} e
\end{aligned}
\tag{5.23}
$$

With the standard state space linear form obtained, the stability around the equilibrium can easily be checked by looking for the eigenvalues of matrix A. For the continues form, the system is said to be asymptotically stable if $\lambda \leq 0$. Towards the digital implementation, the system discretized form is

$$
\begin{aligned}
u(k+1) &= A_{\mathrm{d}} u(k) + B_{\mathrm{d}} e(k) \\
i &= C_{\mathrm{d}} u(k) + D_{\mathrm{d}} e(k)
\end{aligned}
\tag{5.24}
$$

Where index d stands for the discretized version. Finally, to obtain the stability criterion for the discretized system, the eigenvalues λ_k must lie within the unit circle. The eigenvalues can be obtained by satisfying

$$
\det(A_{\mathrm{d}} - \lambda_k I) \tag{5.25}
$$

From 5.25 it is found that the eigenvalues are located at

$$
\lambda_k = \begin{bmatrix} 0 \\ (1-f'_{1,0})\Delta t \\ (1-f'_{2,0})\Delta t \\ \vdots \\ (1-f'_{k,0})\Delta t \end{bmatrix} \tag{5.26}
$$

By adding a constrain to f such that the unit circle perimeter is never reached it follows:

$$
|(1-f'_{k,0})\Delta t| < 1 \tag{5.27}
$$

Solving the inequality yields

$$
\begin{aligned}
-1 < (1-f'_{k,0})\Delta t < 1 \\
-2 < (-f'_{k,0})\Delta t < 0 \\
\frac{-2}{\Delta t} \leq -f'_{k,0} < 0 \\
0 < f'_{k,0} < \frac{2}{\Delta t}
\end{aligned}
\tag{5.28}
$$

Finally, from 5.28 it can be deduced that asymptotic stability for the discrete system requires f_k to be limited in the following way

$$f_{k,\text{lim}} = \begin{cases} f_k(x^+) + \frac{2}{\Delta t}(x - x^+) & \text{for } x > x^+ \\ f_k(x) & \text{for } x^- > x > x_+ \\ f_k(x^-) + \frac{2}{\Delta t}(x - x^+) & \text{for } x < x^- \end{cases} \tag{5.29}$$

with x^- and x^+ the minimum and maximum values of the set of solutions of the equation $f'(x) = \frac{2}{\Delta t}$.

This criterion is enough to always generate a compensator by keeping the system responsive and avoiding too steep decrease of the *RC* time. Although, towards a stronger notion for global stability of a non-linear system, checking for Lyapunov stability is necessary.

The claim of [9] towards equation 5.29 is that when $f_k(x)$ for $x^- > x > x^+$ then the system can become unstable, although such instability will eventually drive the system either towards $x > x^+$ or $x < x^-$ where the criterion is active and the system will effectively stabilize.

5.3.6 Various excitations

Since a direct compensation method is followed, the inverse/backward model does not need to be derived. Again from the two data sets, the first is used to train the compensator and optimize the weights.

The optimization towards this method is done with the `patternsearch` algorithm of the MATLAB Optimization Toolbox [35], and a single cell ($N = 1$) model is used for all excitations.

Then the trained model is applied to the second set.

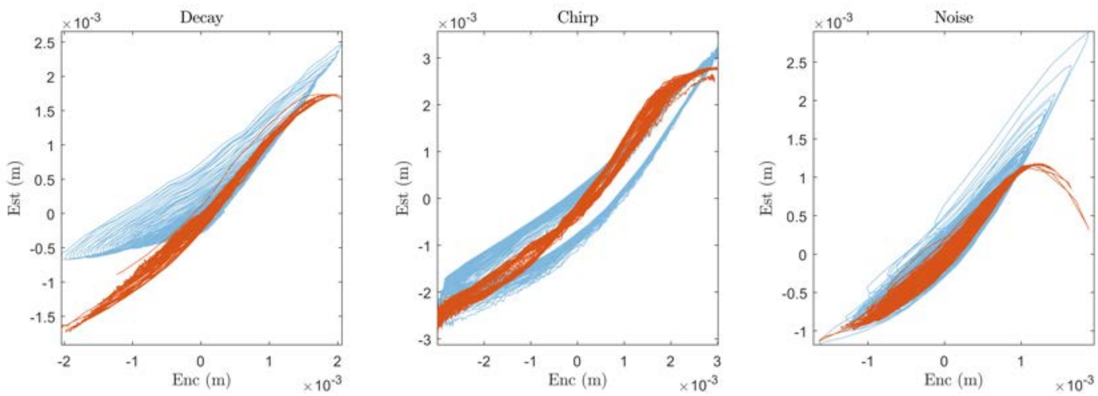


Figure 5.14: Direct compensation fit (orange) over linear fit (faded blue).

In contrast to the mPL of the sine excitation, due to the second non-linear resistor of the sub-circuit (b) the single cell model results with 11 parameters to be estimated instead of 5. The linear fit shown in Fig. 5.14 is obtained with the `polyfit` function of MATLAB which utilizes a least-square estimation algorithm. This fit is used as a comparison measure to the compensator output.

Last, the coefficient of determination R^2 is used as performance criterion and is shown in table. 5.5.

Table 5.5: Ordinary coefficients of determination.

	Model (mPL)	Linear fit (`polyfit`)
Sine	0.990	0.928
Chirp	0.984	0.945
Decay	0.987	0.710
Noise	0.958	0.882

Compensation for all these signals has been applied using both an unconstrained polynomial f_k (eqn. 5.16) and the limited f_k (eqn. 5.29) obtained through the stability criterion. In this case the limited f_k prove to be essential towards fitting of the chirp.

For further details towards initial fitting conditions and optimized weights see [9].

5.4 Conclusion

The Power-law model has been tested (HysTool toolbox) and then developed from scratch by simplifying its structure in terms of discarding (in the sine-wave excitation case) the second non-linear resistor as well as assuming a continues representation based on a hyperbolic sine function. The modified model is analysed, digitally implemented and an identification procedure is followed by means of the Optimization Toolbox of MATLAB and the sum-square error between the real and model output as the minimizing function.

The fit proves sufficiently close to processed measured data. Additionally, the work towards various excitation is briefly described and the relevant modifications are discussed, further validating the model's capability of capturing the hysteretic nature of the sensor response.

Towards derivation of the so-called compensator, the inverse model is described as well as the alternative concept of fitting the forward model to the inverted training data. In the simple case of sine-wave excitation, a direct inversion of the model as described in section 5.3.4 proves sufficient, although, in more complicated excitation the compensator tends to become unstable.

A stability criterion is then developed and described for the alternative model where polynomial representations for the non-linear resistors is assumed. The criterion applies a limit to the polynomial function from which the resistor response is linearized, avoiding the steep decrease of the RC time. The criterion guarantees that, even over longer time, the state of the cell (u_k) will eventually be stable. When applied to the alternative PL, the model strongly linearizes the response of a decaying sine-wave, a frequency sweep and a pure Gaussian noise signal.

Overall, with the proposed modifications, the Power-Law model proves to effectively model and (partially) compensate for the hysteretic non-linearity observed in 3D printed sensors when plotted against the reference signal. As far as the creep modelling goes, further experiments with longer excitations times are required as means to validate whether the modified model is still capable. And if it is, additional analysis on the specific components affecting the creep capturing behavior is crucial.

It is also clear that the compensator output is not a perfectly linearized response. Therefore, the conclusion on whether applying the inverse model directly after the sensor is sufficient for a compensation in a control scheme, or if the modelling errors are detrimental for the closed-loop performance, is still to be evaluated.

6 FEM Based Feedback

6.1 Introduction

The set final goal of the current work is to apply the proposed non-linearity compensation scheme in a real experimental setup. The simplest case to obtain insightful results towards a hardware implementation, is to use the closed-loop already present for gathering the experimental data, but this time using the 3D printed sensor as feedback medium instead of the incremental encoder of the SMAC actuator.

Unfortunately, due to unexpected global circumstances at the time of implementation (Covid-19) and the applied restrictions, access to the laboratory facilities has been restricted. Leading to goal shaping and adjustments such that the work can be continued in a digital manner.

Following the aforementioned circumstances, the decision was made to perform a digital experiment by utilizing the already obtained data as much as possible. The goal is then extended such that the additional step of Finite-Element Analysis (FEA) is introduced as a medium to simulate the sensor behavior digitally.

6.2 Strategy

6.2.1 Goal

The idea behind FEM sensor substitution is that a co-simulation strategy between the control-plant (modeled) in MATLAB and the sensor finite element model (FEA software such as Ansys, Comsol etc.) will provide an approximate non-linear response. Then the compensation via the derived compensator will take place in MATLAB. Fig. 6.1 provides an overview of this idea where Ansys is used as the FEA tool for (1) training the modified PL model (Fig. 6.1(a) in Matlab using the FEA output) and (2) act as sensor simulation tool (Fig. 6.1(b)) in order to generate the approximate '*real*' sensor response during simulation.

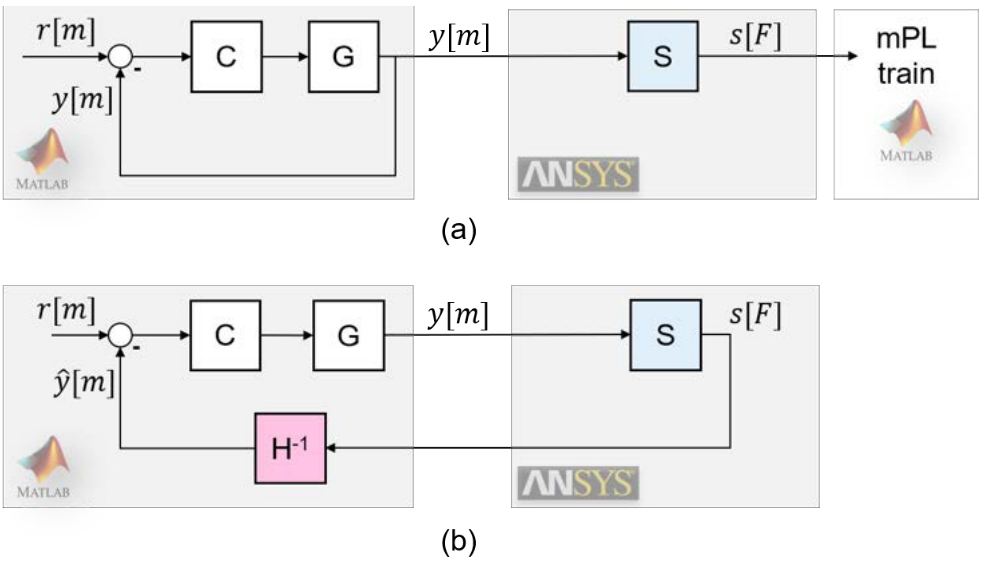


Figure 6.1: Ansys in a loop concept scheme. (a) overview of data gathering setup in digital form, (b) in-loop Ansys simulated response.

6.2.2 Requirements

Of course there are numerous implementation challenges towards the realization of the above schemes. First and foremost, an Ansys version of the sensors is required. Assuming that a successful description of the sensor is present within Ansys, then the next technical challenge is to establish a real-time co-simulation connection between MATLAB and Ansys.

Towards the first challenge, in Ansys a model of the sensors requires:

- 3D CAD model.
- Material mechanical properties.
- Material electrical properties (in case of an eletrical domain analysis).

The 3D model is required for the fabrication of the sensor and therefore is available. The material properties to characterize the sensor such as the proper response are computed from the finite-element engine. Meaning, that the less accurate the material data are, the less realistic the response will be. On the other hand, while setting-up a proof of concept initial scheme, a perfectly accurate model is not the goal. The important objective is for Ansys to generate a model that can mimic the response of the measured data in a satisfactory manner. Then, the response of this model shall be used as means to generate new data to fit the modified PL onto and derive the appropriate compensator.

The reasoning behind developing yet another model (Ansys based) lies in the fact that the fitting process (subsequently the compensator) comes with certain errors. Therefore, it is important for a more realistic simulation to not use the exact same model for generating the sensor behavior and for compensating for it, since this will result in an identity that produces unrealistic ideal results. Furthermore, successfully generating an Ansys description of the sensor, can help systemize the characterization of said sensor via FEA (such that the same procedure may be applied to more materials) and provide additional insight on the mechanical behavior. Worth noting is the fact that current FEA solutions do not support layered structures as produced by additive manufacturing with plastics, but consider most solids as isotropic. This fact alone, limits the accuracy of the response even if a perfectly described material with a well configured solution is assumed.

6.2.3 Material Model Options

The main challenge in this phase, is to describe the non-linear response of the given material (eTPU) within Ansys. From the pre-defined materials within the Ansys libraries, the following options are available:

- Linear models
- Non-linear models
- Non-linear models (specialized)

The linear models naturally do not suit the given case. However, a number of non-linear material descriptions are included (e.g. hyperelastic models), which add a layer of complexity in terms of curve fitting or knowing the material constants beforehand and additional solution configuration. Further inspection of the Ansys libraries, reveals that the a proper candidate viscoelastic model capable of exhibiting rate-depended hysteretic behavior, is the so-called Bergstrom-Boyce model (BB) which is under the category of specialized hyperelastic models (no curve fitting support). Additional sources (explained later in 6.2.4) provide more options towards modelling viscoplastic polymers such as the Three-Network Viscoplastic model (TNV) [8].

6.2.4 External model calibration

For training several non-linear material models, the MCalibration package from PolymerFEM software suite is used. There are several ways to train-calibrate a material model with the MCalibration. All of the options can be described in two categories:

- Pre-set Cases.

  The software provides the option to use common pre-defined load-cases which becomes useful when data from specific well defined experiments are available, such as stress-strain data, creep data, stress relaxation data, friction tests, fatigue tests, Poisson's ratio, E-modulus, to name a few. In [48] a systematic procedure on obtaining the proper experimental data for the case of a BB model is described.

- Custom load-cases.

  Configuring a custom load-case requires an external solver (such as the one from Ansys) for solving an inverse problem. Includes generic cases with Force-Displacement data.

6.2.5 Adaptation

Measured quantity issue

The problem that arises at this point, is related to the fact described under chapter 3 where the obtained data are analysed. The current measurement data do not fit any of the pre-defined categories (displacement-voltage measurements).

Assumption

Immediate deduction is that the measurements are not compatible neither meant to be used with the material models such as BB or TNV. Of course, that was known and explained in section 3. Nonetheless, following the decision that an accurate description of the sensor in terms of mechanical characterization is not the focus, and additionally, following the aim to shape the given model such that response from Ansys fits the sensor, an attempt can be made to curve-fit either the BB or TNV model to the data. A custom load-case can be defined for a Force-Displacement experiment where the force is replaced with the voltage measurements, assuming of course a linear relationship between force and voltage.

A calibrated model then, given the same excitation (sine-wave displacement), is expected to generate a Force-Displacement (at a node) output similar to the original Voltage-Displacement measurement. The new simulated results will then be used to derive a new compensator in MATLAB.

6.3 Non-linear material models

Towards the two non-linear models, BB and TNV, following is a brief introduction to said models. Since, as can be seen later, the implementation of the models only requires the use of experimental data during the calibration phase, an analytical description of the derivation of these models is not given, instead a descriptive overview can be found in [8].

6.3.1 Bergstrom-Boyce (BB)

The BB is an advanced model for predicting the large-strain, time-dependent behavior of elastomer like materials [8]. Its rheological representation consists of two spring and a damper configuration in parallel as shown in Fig. 6.2.

For the fabricated sensor, as explained in chapter 3, two polymers are used. A common TPU for the sensor body and a conductive variant of a TPU polymer for the strain gauges. From comparison of the BB model parameters (PolyUMod manual (MCalibration) [8]) and the material properties of the used TPU [39] it is concluded that several parameters are unknown/unavail-

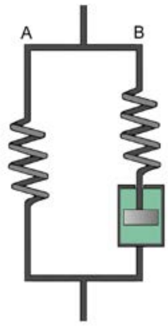


Figure 6.2: Rheological representation of the Bergstrom-Boyce model [8].

able towards an initial estimate of the BB model. Fortunately, there is another solution towards obtaining the remaining properties which relies on the curve-fitting solution using experimental data.

The BB model is the best candidate model available within the Ansys library, although is not specifically designed to model thermoplastics but focused towards rubber material. It can be used for modeling said thermoplastics but the yield evolution will not be properly captured [49].

6.3.2 Three-Network Viscoplastic (TNV)

Looking at third party packages that are able to generate Ansys compatible material models (details in sub-section 6.3.3), the TNV model is the suggested solution towards modelling thermoplastic materials.

The TNV model is presented as a general purpose viscoplastic model capable of capturing the experimentally observed behaviors of thermoplastic materials. These behaviors include time-dependence, pressure-dependence of plastic flow, pressure dependent bulk modulus, volumetric plastic flow, damage accumulation, and triaxiality dependent failure [8].

Briefly, the model consists of three parallel networks (see Fig. 6.3). Each network can be thought of as the combination of a non-linear spring in series with a non-linear damper. The TNV model requires a considerable larger number of parameters. Some of the features can be deactivated by purposely setting certain parameters to a zero value. For the parameters related to the activated features, an iterative identification process for estimation is used given a set of experimental data.

6.3.3 MCalibration

The BB and especially the TNV model comes with increased complexity. Additionally, since the curve-fitting feature is not supported for the Ansys implementation of BB model, as far as Ansys is concerned the material parameters need to be provided. Fortunately, the author of the BB model offers an external package called *PolymerFEM* that includes the tool *MCalibration* used to auto-calibrate a wide variety of non-linear models, including the BB and TNV, given a set of experimentally obtained measurement data, as explained in section 6.2.4. Given the measured quantity issue, the only viable option is the route of custom-load case where an inverse calibration takes place.

Calibration process overview

The steps required for a calibration cycle are:

- A complete solution configuration in Ansys given a default (linear) material.

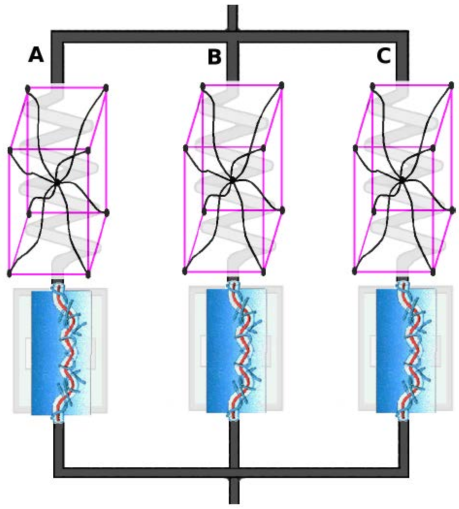


Figure 6.3: The TNV material model framework that consists of three parallel networks [8].

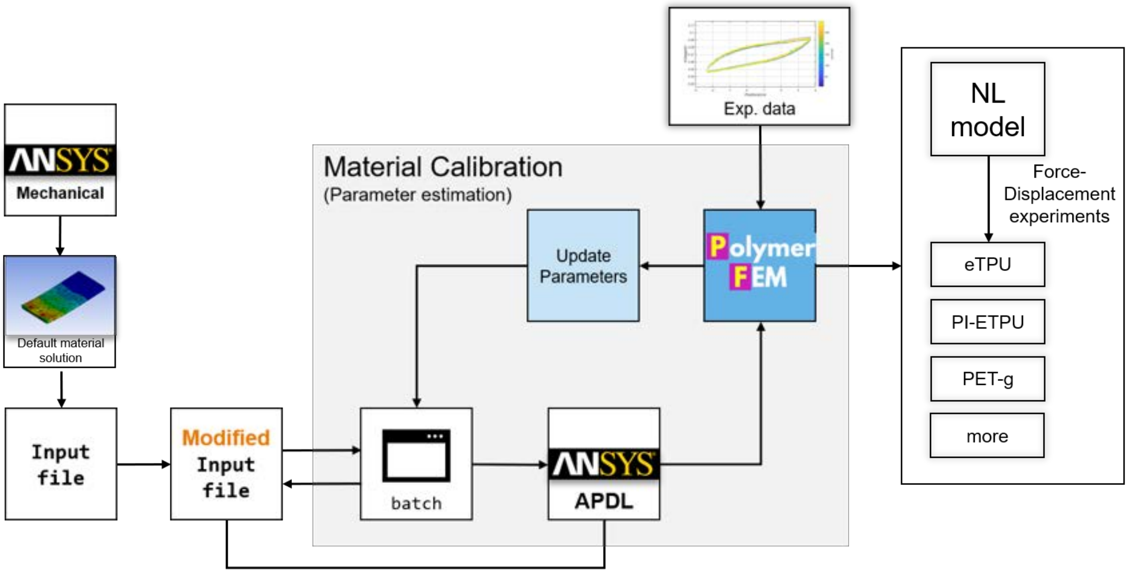


Figure 6.4: Model calibration scheme via MCalibration.

At this state, it is most convenient to work with the user interface of Ansys (Ansys Mechanical) for a more intuitive setup of the simulation.

- Generation of the Input-file carrying the complete configuration in APDL commands.

  Once the solution is ready, the entire configuration can be exported in auto-generated scripting commands.

- Input-file modification such that the desired property (Force,stress etc) is extracted from a certain node and written to a given location.

  The most crucial step in the current implementation is the extraction of the proper parameter. This requires user intervention in the input-file. First, the default material is replaced with a link to an external file that is iteratively modified by MCalibration. Second, an algorithm must be set during the post-processor phase of the simulation responsible for extracting the desired Force, stress or strain from a specific node at each simulation sub-step. The extracted values should then be stored for reading from the MCalibration package.

- MCalibration material selection and optimization approach configuration.

  The chosen material models in this case are the BB and/or TNV. To start with less cpu load, the minimum amount of necessary parameters is used for the optimization. There are several algorithm options available to choose for the optimization approach, such as *Random Search, Levenberg-Marquardt, Quasi-Newton, Global Optimum Search* and others.

- A batch file (Windows) that guides MCalibration to the Ansys solver, initialize the *batch mode* operation of Ansys core (APDL) and executes the modified input-file.

  At this phase an iterative process starts that involves the optimization of the set material in MCalibration before passing at each iteration the current parameters in the input-file.

At this state, the configuration required for Ansys to simulate the sensor, given the part geometry and a proper material model, is layed-out. As a last step before proceeding into an implementation attempt, the communication between the two environments (Matlab-Ansys) should be established. This process is described in appendix section 9.9.

6.4 Initial FEM solution

A basic FEA solution with a default linear material is the first implementation step. The analysis is based on a simple beam loading configuration as shown in Fig. 6.5. From [39] the material properties of TPU are used to generate the default material (linear). The simulation is set for load-unload cyclic excitation via a sinusoidal signal at 0.5 Hz.

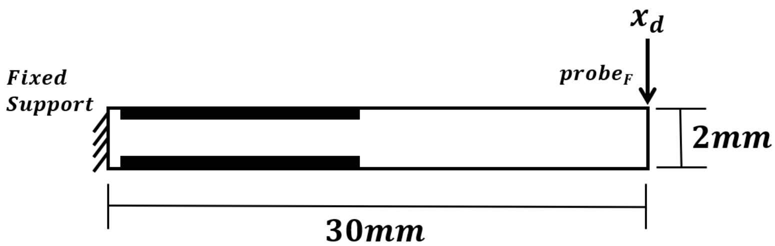


Figure 6.5: Beam loading at SMAC stroke point.

The specific details of the analysis and solutions settings are provided in the Appendix 9.9.

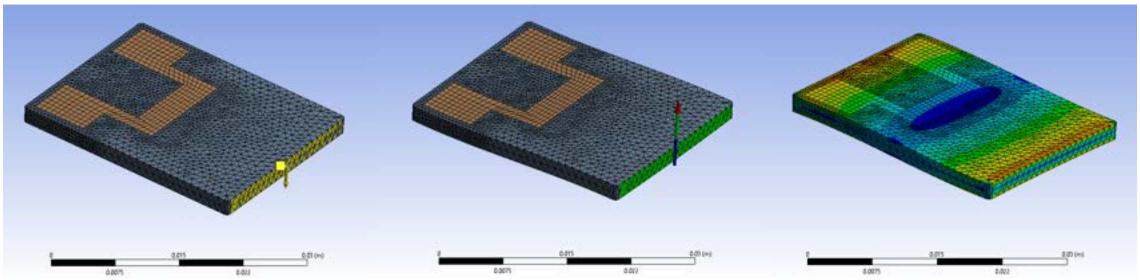


Figure 6.6: Initial FEA simulation. Displacement load (left), Reaction force (center) and equivalent stress (right).

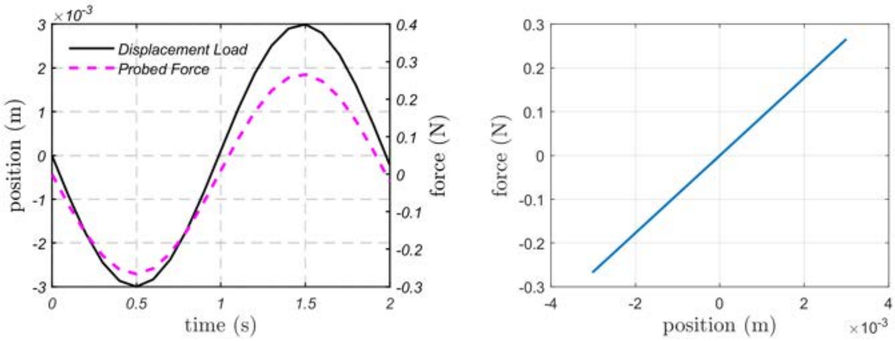


Figure 6.7: Ansys simulation with linear material model for one cycle of loading-unloading. Probed Force represenets the reaction force at the node where the displacement is applied.

Looking at Fig. 6.7 the effect of the linear material is clear. This is the expected behavior. The entire configuration can now be exported in an Input-file to be used with Ansys APDL and MCalibration software.

6.5 Model calibration

Second phase of the strategy is the calibration of the non-linear BB or TNV model using the MCalibration software. With the Input-file ready, the two crucial user interventions are (1) to replace the material definitions within the Input-file with pointers to *tuning* materials from MCalibration and (2) implementing a data extraction and storing algorithm during the post-processor phase of the FEA solver. The flow chart[1] of Fig. 6.10 describes the actions for (1) and (2).

The pipeline for the communiction between Ansys APDL and MCalibration throught the batch file that calls the modified Input-file is successfully working and an iterative optimization is possible.

The processed data consist of one loading-unloading cycle as shown in Fig. 6.8 and 6.9.

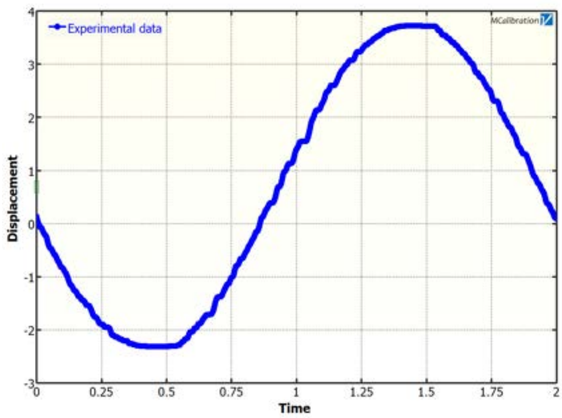


Figure 6.8: Model output for initial conditions.

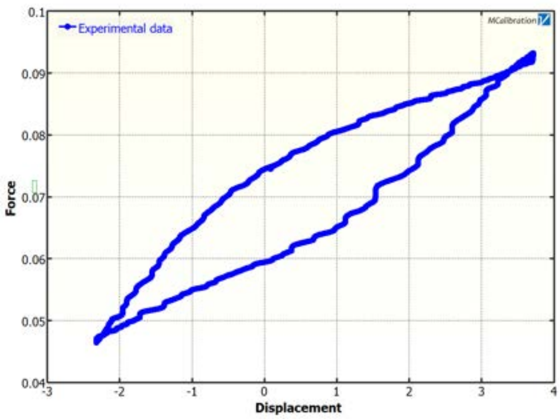


Figure 6.9: Model output with fitted weights.

Unfortunately, a good fitness performance was impossible to obtain.

There are several possible reasons behind the fit failure. Based on the failed results these may be:

- Poor initial conditions for the optimization problem.

[1]Code based on example provided in the tool documentation.

- Returning force prediction from the FEA inverse solution too low in magnitude.
- Data extraction algorithm implementation is prone to errors.
- Voltage output of the experimental sensor may be a bad indicator of loading force.

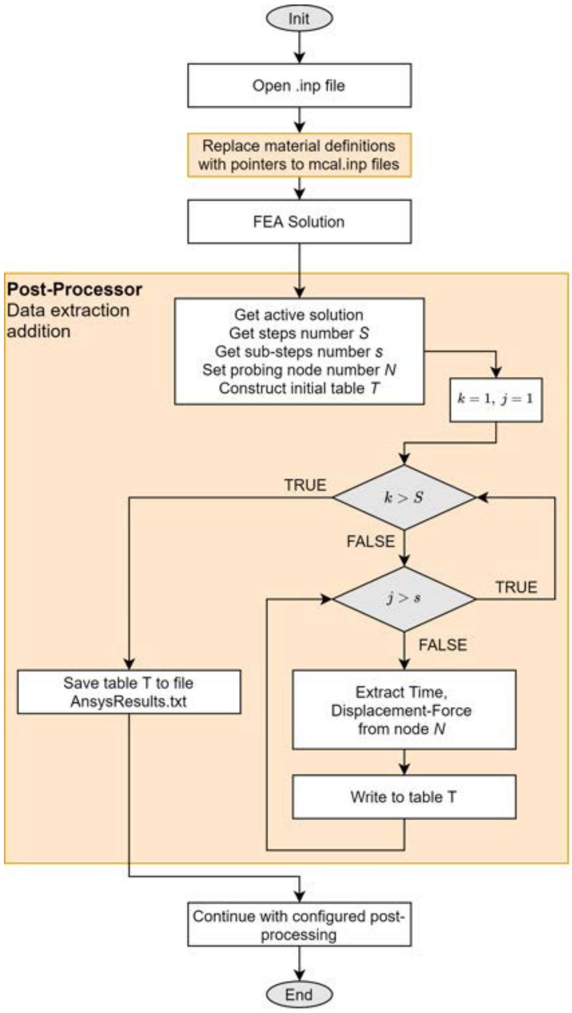


Figure 6.10: Input file required modification for material tuning through MCalibration. Orange area represents the points of user intervention. Actual code can be found in Appendix sec. 9.6.

Troubleshooting the given scenario has a high complexity due to the measured quantity issue leading to the lost meaning of the parameters which effectively limits the effectiveness of the initial guess through intuition. Additionally, towards the TNV model, the latter is magnified because of the large number of estimation parameters.

6.6 Conclusion

Overall, it is concluded that in order to obtain a non-linear material description in Ansys either the material constants need to be defined beforehand, or curve fitting process (Ansys native or not) should take place as means to obtain the parameters.

Given the *gray area* of the entire FEM-based sensor scenario (Fig. 6.4) is the fact that the appropriate data are not present. And additionally, the limitation towards a dual-simulation, where Matlab can only provide inputs for FEM once and wait until the results are generated (custom implementation), instead of a true co-simulation where Matlab has access to the running

FEA solver substeps (official toolbox) [further explanation in appendix 9.9], there is not enough confidence to continue in the path of this strategy.

Although, the overall description of the process towards calibrating a non-linear model (e.g BB or TNV) is believed to be sound when provided with the appropriate experimental data (such as stress-strain, or any data that would fit the pre-set load-cases, see sec. 6.2.4). Then the generated models should exhibit the mechanical domain non-linearities observed in the sensor, and could be used for a FEM analysis with higher confidence.

7 Control Strategies

7.1 Introduction

From chapter 6 it is clear that, given the lack of sufficient data, the FEA strategy based scheme of Fig. 6.1 cannot be applied. Nevertheless, following the work of [9] where several modifications of the PL model are presented, there is the possibility to implement the entire simulation within the MATLAB Simulink environment by employing one of the models as the approximation of the *real* sensor. That is, the complexity from the non-linear material model calibration as well as the dual-simulation scheme is discarded in favor of obtaining insight in the control scheme, omitting any characterization towards the mechanical properties of the actual sensor.

The discussions and insights from the following strategies are based on analysis from the perspective of the simulation results only. Subsequently, the goal of the proposed schemes is to provide an overview of possible approaches towards compensation. Based on the simulations, insight on the compensation performance and limitations is drawn in the absence of a thorough stability analysis.

7.2 Feedback Position Control with Output Estimator Compensation

The idea behind the only-MATLAB approach is similar to the one presented in Fig. 6.1 but the Ansys modelled sensor is replaced with one of the models developed in [9]. As explained in chapter 4 in that work four different experiments were carried out, one for each excitation (sine,decaying sine, sine frequency sweep and random signal (noise)). From the results of these four excitations, it can be deducted that the noise-trained model is the best candidate to represent the *real* sensor due to the wider range of frequencies and amplitudes included in the excitation. Therefore, from here on, the noise excited response will be used to model the *real* sensor in the loop, instead of the Ansys generated response.

Analysis

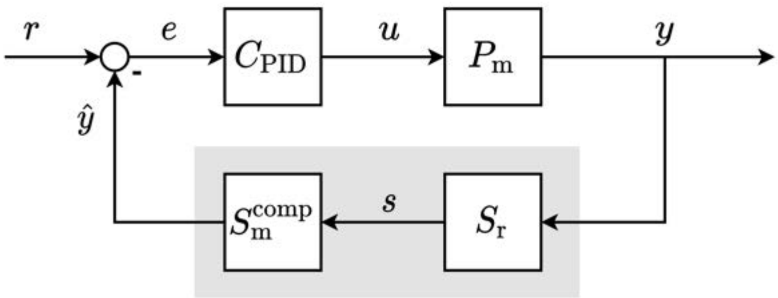


Figure 7.1: Feedback loop with direct application of the compensator.

Fig. 7.1 describes the feedback loop where the estimated position $\hat{y}$ is used to calculate the driving error $e(t) = r(t) - \hat{y}(t)$ fed into the controller C_{PID}. The plant P_{m} here represents the SMAC stroke which can be modelled by a simple moving-mass system since the friction off the bearings and the stiffness of the sensor are expected to be much smaller than the force needed to accelerate the mass. Therefore

$$P_{\text{m}} = \frac{1}{m_{\text{stroke}} \cdot s^2}, \quad \text{with } m_{\text{stroke}} = 0.250[\text{Kg}] \tag{7.1}$$

S_{r} stands for the *real* sensor representation, in this case modelled as the mPL noise version model with output equation 5.17 using the estimated weights that fit the original measurements from the lab experiments (Fig. 7.2). $S_{\mathrm{m}}^{\mathrm{comp}}$ represents the derived compensator from the first modification of the original PL model where a hyperbolic function is used to describe the non-linear resistor for sub-circuit (a) (see ch. 5). Additionally, for a better fit, the $S_{\mathrm{m}}^{\mathrm{comp}}$ model is extended with a second non-linear resistor at the output, represented by a 3$^{\mathrm{rd}}$ order polynomial of the form described by 5.16.

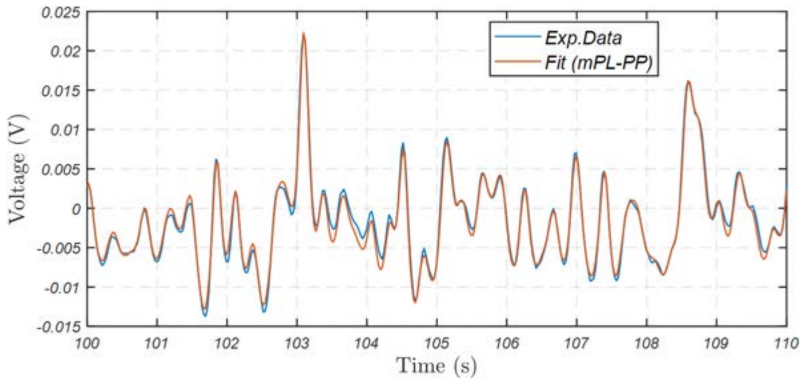
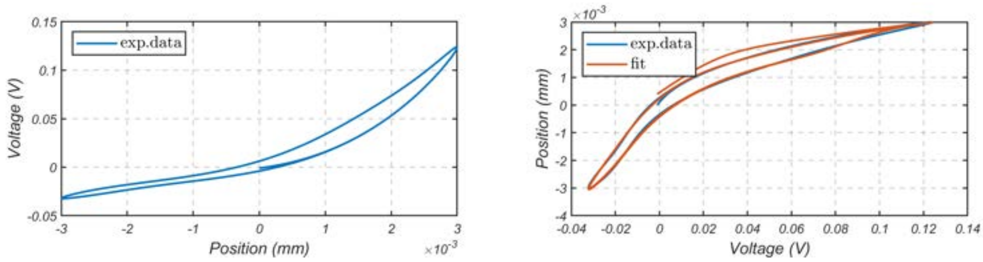


Figure 7.2: Noise excitation experimental data used for training the *real* sensor representation. Coefficient of determination $R^2 = 0.9793$. PP: polynomial representations for both non-linear resistors of the circuit [9].

To gather data for the parameter estimation a simulation is run where the loop is closed with plant output y. The logged time t, reference r and sensor response s are used to directly derive the compensator with s input and $\hat{y}$ output. Here, S_{r} is excited with a sine-wave of frequency f =0.5 Hz and amplitude A =3 mm. Figs. 7.3a and 7.3b show the measured response as well as the inverted data and fit.



(a) Noise fitted mPL response to sine excitation.

(b) Inverted simulation data & fit with fitness measure $R^2 = 0.995$.

Figure 7.3: mPL compensator training towards simulation.

The controller C_{PID}, is a discrete in-series PID controller as seen in Eqn. 7.2, with time-step $t_s = 5.04 \cdot 10^{-4}$ derived from the filtered measurement data.

$$C_{\mathrm{PID}} = K_{\mathrm{p}} \cdot \left(1 + \frac{1}{T_{\mathrm{i}}} \cdot \frac{T_{\mathrm{s}}}{z-1} + \cdot T_{\mathrm{d}} \cdot \frac{z-1}{T_{\mathrm{s}} z}\right) \tag{7.2}$$

Towards tuning, the process followed is next explained. First a controller is tuned for the linear closed-loop system $H_{\mathrm{L}} = C_{\mathrm{PID}} P_{\mathrm{m}} / (1 + C_{\mathrm{PID}} P_{\mathrm{m}})$ with [50] in the frequency domain and a target cross-over f_c =20 Hz. Then, concerning the complete non-linear system, starting with the derived C as a basis, than plan is to follow an iterative process where, based on the response of the system, first the proportional gain is adjusted either to reduce overshoot by reducing the gain or make the system faster by adding gain. The integral time is adjusted with the aim to eliminate the residual steady-state error. The derivative time is then modified as means to improve the system settling time. It should be noted that starting with a controller tuned for the linear system is expected to yield sub-optimal results. In the current scenario, under the assumption that the compensator is compensating for the majority of the non-linearities (following the results of ch. 5), in case only weak non-linearities remain the system, the controller is expected to still be operational at relatively low bandwidths. Initiating the tuning process, it is found that the controller tuned for the linear system, sufficiently operates. Therefore there is no need to enter the phase of iterative tuning based on the system response.

The tuned paramaters are K_p = 545 the proportional gain, T_i =0.029 s the integration time, derivative time T_d =0.0573 s and $T_\mathrm{s} = t_\mathrm{s}$.

The open-loop system with output y is described with a bandwidth of f_c =20 Hz (Fig. 7.4).

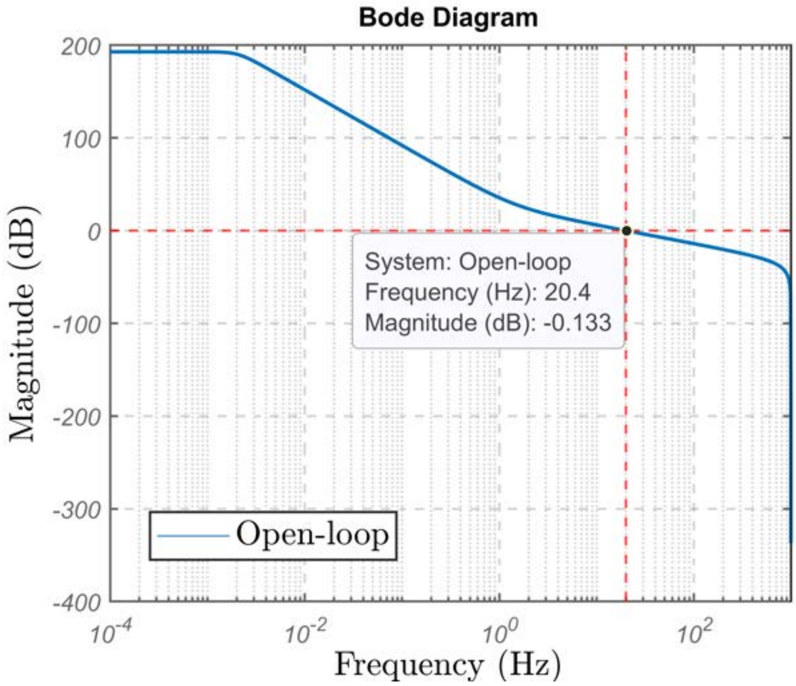


Figure 7.4: Bode magnitude plot of the linearized system at $t = 0$.

The reference signal is initially set to be identical to the excitation used towards obtaining the compensator, therefore:

$$r(t) = 0.003 \cdot \sin(\pi t) \tag{7.3}$$

s describes the sensor response, which is the compensator $S_\mathrm{m}^\mathrm{comp}$ input.

The transfer from r to y is then described with (assuming $S_\mathrm{m}^\mathrm{comp}$ perfectly linearizes S_r)

$$H_{y/r} = \frac{P_\mathrm{m} C}{1 + P_\mathrm{m} C S_\mathrm{m}^\mathrm{comp} S_r} \tag{7.4}$$

In this scenario, the ideal situation would be when the model perfectly fits the real sensor data and no errors are produced during fitting or by inverting the model. In such an ideal case then

$$\hat{y} = \underbrace{(S_\mathrm{m}^\mathrm{comp}(S_r))}_{\approx I} y \tag{7.5}$$

Here the importance of an accurate model becomes clear: since the more the pair $S_{\mathrm{m}}^{\mathrm{comp}} S_r$ strays away from I the worse the plant position estimation is.

The scheme has been implemented in MATLAB Simulink for simulation. It should be noted, that the entire model is re-implemented in Simulink enviroment from scratch, meaning that it is not as thorough tested as the previous derived models. Nonetheless, the response of the Simulink model has been evaluated with the Matlab models and found to be identical in the given scenario. The results of 5 periods are then shown in Fig. 7.5.

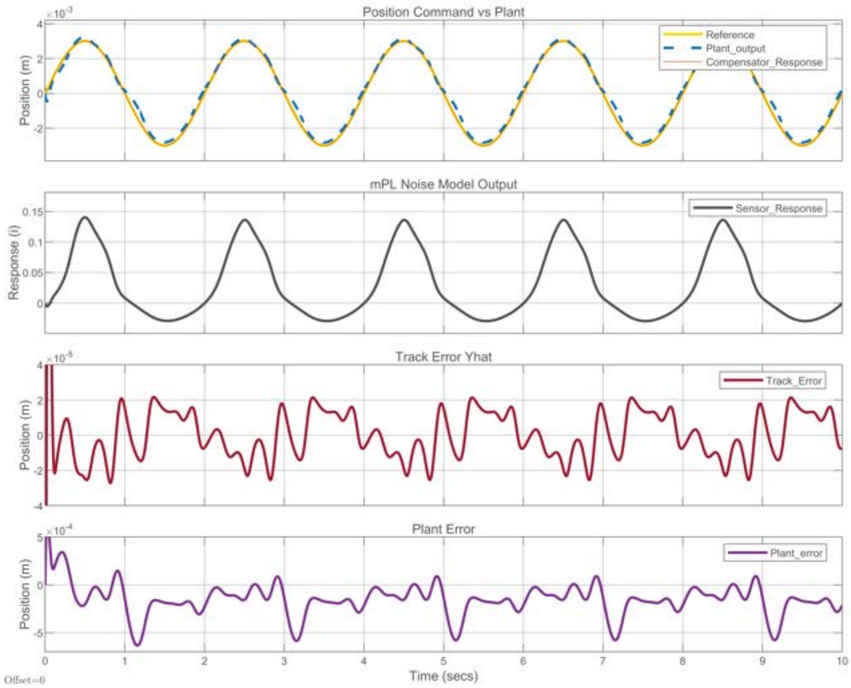


Figure 7.5: Feedback loop with direct application of the compensator. From the top (1) Reference position, Plant positon and Estimated plant position respectively. (2) sensor response, (3) Track error fed into the controller (estimated position from compensator), and (4) Plant error.

Plant error performance:

$$
\begin{aligned}
e_{\mathrm{p}_{\max}} &= 6.94 \cdot 10^{-4} \\
e_{\mathrm{p}_{\mathrm{mean}}} &= -1.543 \cdot 10^{-4}
\end{aligned}
\tag{7.6}
$$

The effect of the fitting errors on the *real* plant position is clear. The tracking error shows a good performance with a average error of $3.93 \cdot 10^{-7}$. This means that the controller does a good job minimizing $e = r - \hat{y}$. On the other hand, it can be seen that even with a very good fit, the plant output still deviates enough to produce a higher average error[1] ($e_{\mathrm{p}_{\mathrm{mean}}}$) and this effect is also visible in the position tracking of Fig. 7.5 (top).

Because the non-linearity is not entirely eliminated, there exists a *counter-effect* when controlling the modelled compensator's output that affects the position of the plant.

Although the scheme shows to adequately follow the reference, the errors of 7.6 reveal a performance that in certain cases may be considered poor in terms of accuracy. Especially given

[1] The minus sign of the mean error is more conveniently understood by visualizing the loop crossing that remains from the un-compensated non-linearities.

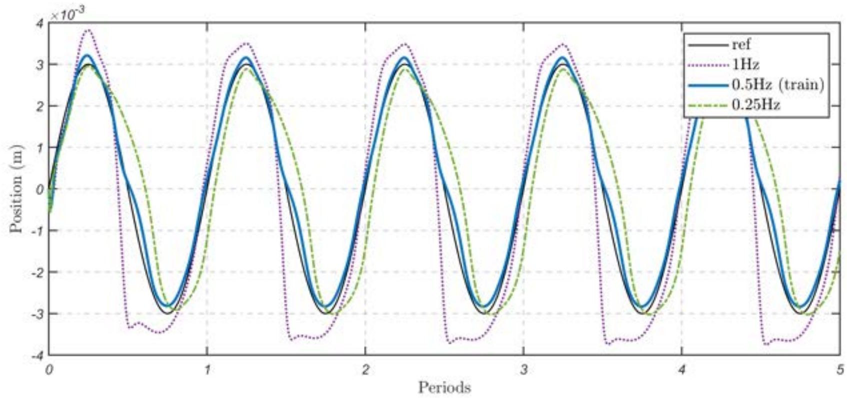


Figure 7.6: Plant position versus excitation on three different frequencies.

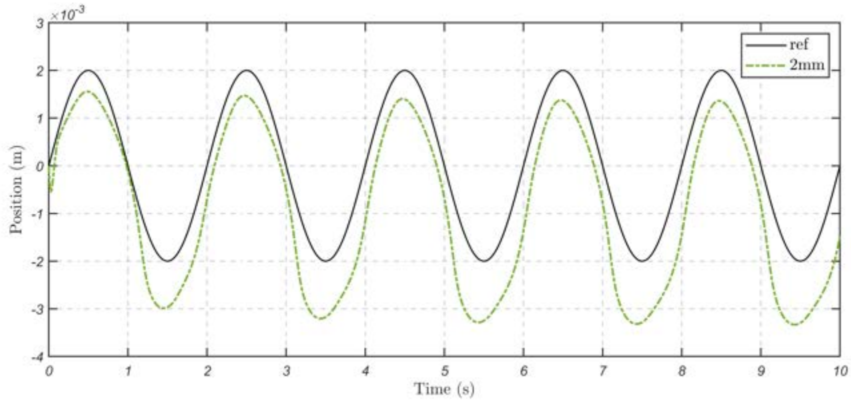


Figure 7.7: Plant position versus excitation on two different amplitudes.

the ideal configuration of the simulation (linear plant, no disturbances, low-frequent excitation). Although, applications that require very low error performance in combination with fast tracking would benefit more from the linear response of traditional sensors.

Towards further testing of the scheme's performance the excitation is modified by lowering and increasing the frequency.

Fig. 7.6 reveals the performance of the plant's output in three different frequencies. It is immediately clear that a change on the trained frequency affects the performance of the system significantly. Decreasing the frequency has a slightly less impact than increasing, although in both cases the position is outside of any acceptable error for most applications.

A similar observation is seen when exciting the plant at a fixed frequency (0.5 Hz) and varying amplitude. Notably, increasing the amplitude ($A > A_{\text{train}}$) has a severe impact on the stability of the system. The compensator becomes unresponsive at the first corner case (output grows to infinity) and subsequently the controller fails to minimize the error. This is not the case for decreasing the amplitude, although the performance is still very poor as seen in 7.7 where an additional drift is revealed.

It is clear that the scheme of Fig. 7.1 appears to be very sensitive in any change of the periodic excitation signal. The source of the performance decrease is the failure of the compensator to properly invert the sensor's non-linearity. The work of [9] towards fitting the noise excited sensor measurements is the most likely model to perform better in this case since this model fits a non-periodic entirely random excitation. Next, an alternative scheme is considered as means to avoid the extra complexity added by the inverse compensator.

7.3 Feedback Indirect-Position Control with Model-Weighted Input

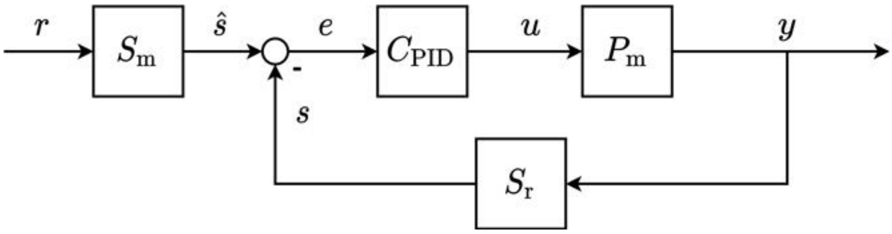


Figure 7.8: Input-Weighting based on the modelled sensor response scheme.

An alternative scheme that discards the need of an inverse model (or inverse data trained forward model) is illustrated in Fig. 7.8. In this case, in contrast to the direct application of the compensator to the feedback loop, the error is not tracking position but sensor response (modelled as current output i of the mPL in this case subtracted from the real measurement which is the filtered differential measurement of the strain gauges voltage response).

S_r, S_m (forward-model) and r are the same ones used in scheme 7.1. Towards the controller, in this case, the system is found to severely over-react given the previously tuned controller. Therefore, entering again in the tuning phase, the most notable change is the significant reduction of the proportional gain as means to avoid the system's overshooting. Subsequently the effective open-loop bandwidth is greatly reduced.

Post-tuning, the parameters are K_p = 16.76 the proportional gain, T_i = 2.46 s the integration time, T_d = 0.4619 s the derivative time and $T_s = t_s$ the sample time.

In this case the open-loop system is described with a cross-over frequency of $f_c \approx 5$ Hz (Fig. 7.9).

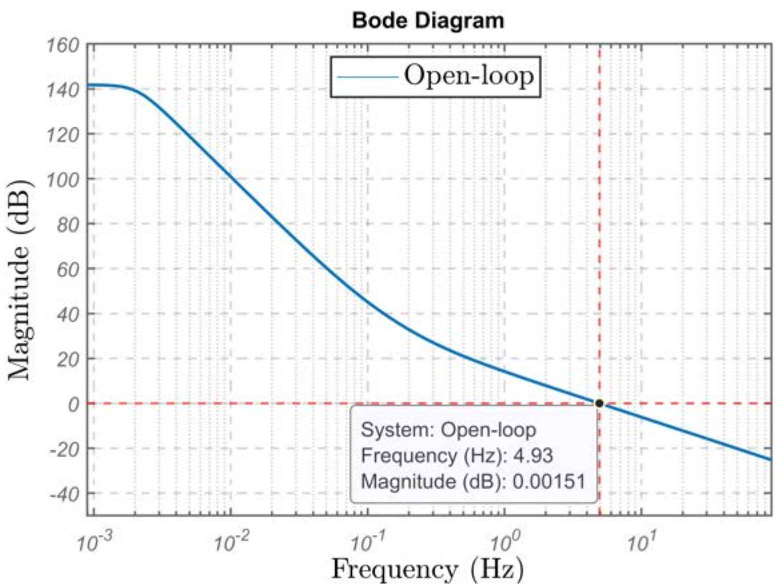


Figure 7.9: Frequency response of the open-loop system with input-weigthing compensation.

The new transfer from input r to output y is then described as follows[2]:

$$H_{y/r} = \frac{P_m C_{PID} S_m}{1 + P_m C_{PID} S_r} \tag{7.7}$$

[2]Again assuming S_m^{comp} perfectly linearizes S_r.

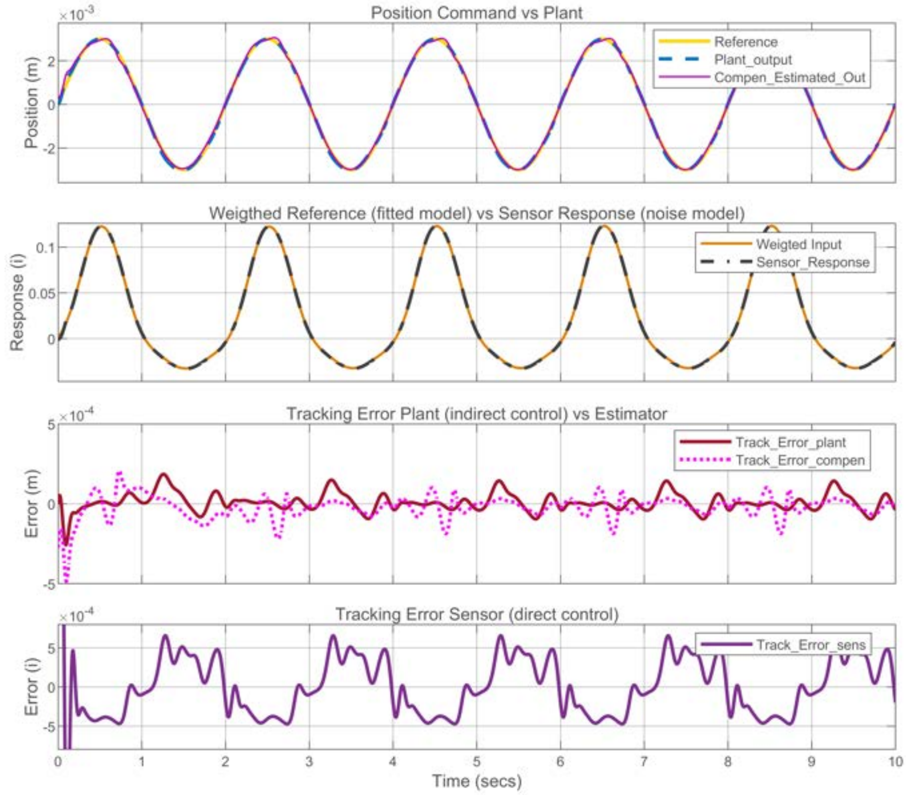


Figure 7.10: Feedback loop with model-based input-weighting compensation.From the top (1) Reference position, Plant positon and Estimated plant position respectively. (2) Weighted reference and sensor response, (3) Error between the reference r with the plant position and estimated position respectively, and (4) Track error fed into the controller (sensor response).

First note from Fig. 7.8 is that the plant position y is not directly part of the tracking but the main goal is still to position y such that $e_{\text{plant}} = r - y$ is minimized, even though the signal y is not available.

The idea is that assuming a good model S_{m} the signal $\hat{s}$ represents the ideal reference that will minimize e_{plant} given the non-linear response of the sensor. Downside is that fitting errors are given as the ideal reference and fed into the tracking, although fitting errors are also a limiting factor in the previous scheme. On the benefit side, the scheme is not depended on model inversion (or inverted data fitting to derive a compensator), which, given the simulation results of Fig. 7.6 under-performs.

At the initial measured frequency and amplitude the performance is comparable to the direct compensation scheme in terms of how the remaining non-linearities counter affect the plant output. Although, with a closer look in the plant errors it is clear that e_{plant} is significantly lower than the direct compensation with the average error as seen in eqn. 7.8.

Plant error performance:

$$\begin{aligned} e_{\text{p}_{\text{max}}} &= 1.86 \cdot 10^{-4} \\ e_{\text{p}_{\text{mean}}} &= -2.29 \cdot 10^{-5} \end{aligned} \tag{7.8}$$

Furthermore, the illustration of Fig. 7.8 does not provide an estimation of the plant output $\hat{y}$ for the user and in realistic situation the original y signal is not measured. To obtain $\hat{y}$, either the previously derived compensator shall be used to obtain an estimate $\hat{y}$ given the sensor response, or a model of the plant can be employed as means to generate $\hat{y}$ given the controller effort. In this case since no identified plant or hardware-in a loop setup is present, the plant is already a model. Therefore, the scheme of Fig. 7.8 can be extended as in Fig. 7.11. The estimated position along with its error can be seen in Fig. 7.10. At the fitted frequency and amplitude, the estimation may be consider satisfactory since the error is of low magnitude (10^{-4}) and the fact that at the current state this estimate does not play any crucial role in the feedback loop.

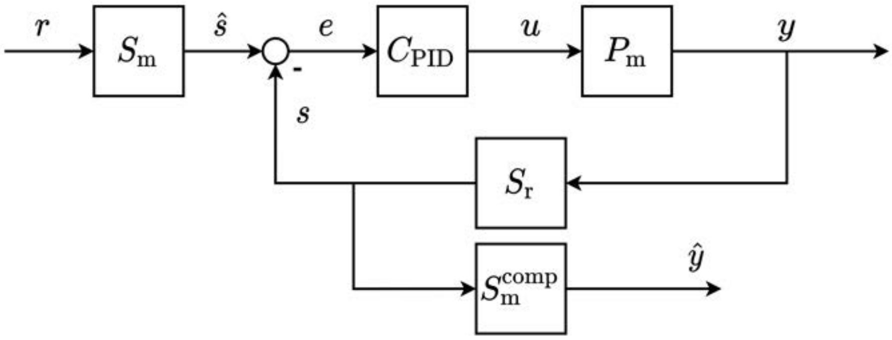


Figure 7.11: Plant estimation extension on 7.8.

From Fig. 7.10 it can seen that the controller does again good job at minimizing $e_{sens} = \hat{s} - s$ and the counter-effect on the plant position is less emphasized with the average plant error one magnitude lower than the previous scheme. Hence, an improvement in performance is already observed in terms of error performance. Next, the same steps towards modifying the excitation are used.

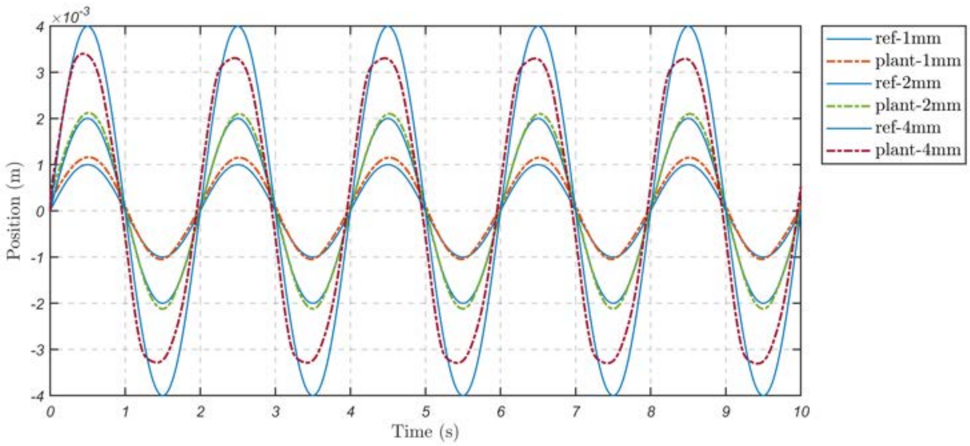


Figure 7.12: Various amplitudes of excitation for input-weighting based scheme. Each case plotted in pair with its reference (blue).

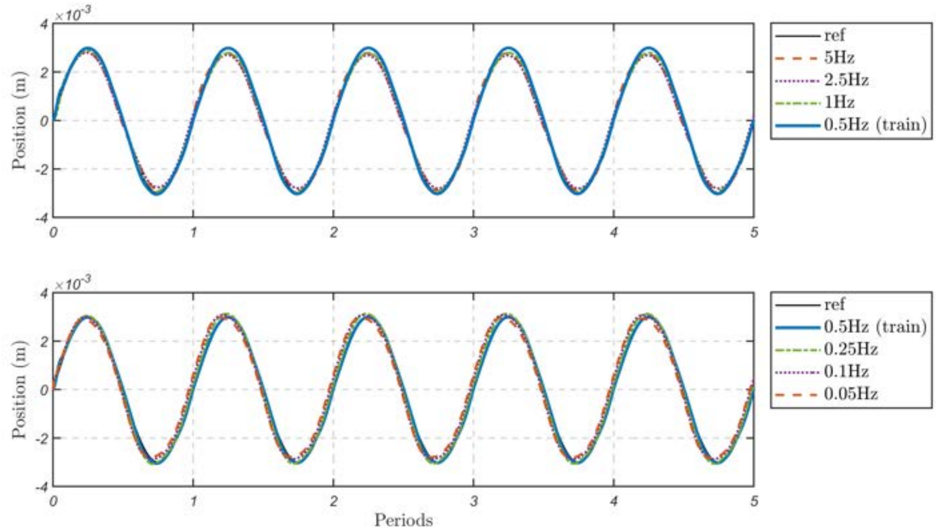


Figure 7.13: Various frequencies of excitation for input-weighting based scheme. Increasing frequency (top), decreasing frequency (bottom).

Looking at the results of figures 7.12 and 7.13 it is clear that the alternative scheme performs better when compared to the direct application of the previous scheme. From Fig. 7.13 it can be seen that this compensation scheme handles the frequency variation significantly better. Amplitudes variation though, have a higher impact on the plant response as seen in Fig. 7.12. Specifically, while decreasing the amplitude shows an adequate tracking, increasing amplitudes start to show a tracking performance decline, where not only the sine-wave minima and maxima are not followed by the plant, but the constant mismatch with the reference reveals a hysteretic response from the plant. This effect only worsens by further increasing the amplitude. Although, again in contrast to the direct compensation scheme where the higher amplitude rendered the system response unstable, the current approach is still able to be responsive and the controller actually works as intended (minimizing e_{sens}).

7.3.1 Third-order trajectory test

For the next test, with the higher confidence obtained from the alternative scheme results, the system is excited with a cubic trajectory as means to evaluate the model behavior given an entirely different excitation signal. The trajectory is constructed given an arbitrary set velocity and acceleration boundaries with the help of `cubicpolytraj` [51]. The boundaries for acceleration and velocity are $a_{\mathrm{b}} = |0.15|$ m/s^2 and $v_{\mathrm{b}} = |7.5 \cdot 10^{-3}|$ m/s respectively. The jerk j_k is obtained from the acceleration slope. The position waypoints x_{p} can then be computed through integration (Eqn. 7.9).

$$x_{\mathrm{p}} = \sum_{k=1}^{n} \left(\frac{j_k}{6} \cdot t_{\mathrm{i}}^3 + \frac{a_{k-1}}{2} \cdot t_{\mathrm{i}}^2 + v_{k-1} + x_{k-1} \right) \tag{7.9}$$

Where n is the number of points present in the profile and t_{i} the time interval between two points. The resulting profile is shown in Fig. 7.14.

Overall the tracking of the cubic trajectory is good, first observation would be the compensator output (for the estimated $\hat{y}$) is not capturing the y, once again hinting that a compensator trained at a fixed frequency and amplitude is inadequate. Since the estimation of y is not crucial at this scheme, for now the focus is shifted on the *actual* plant performance. Fig. 7.15 shows

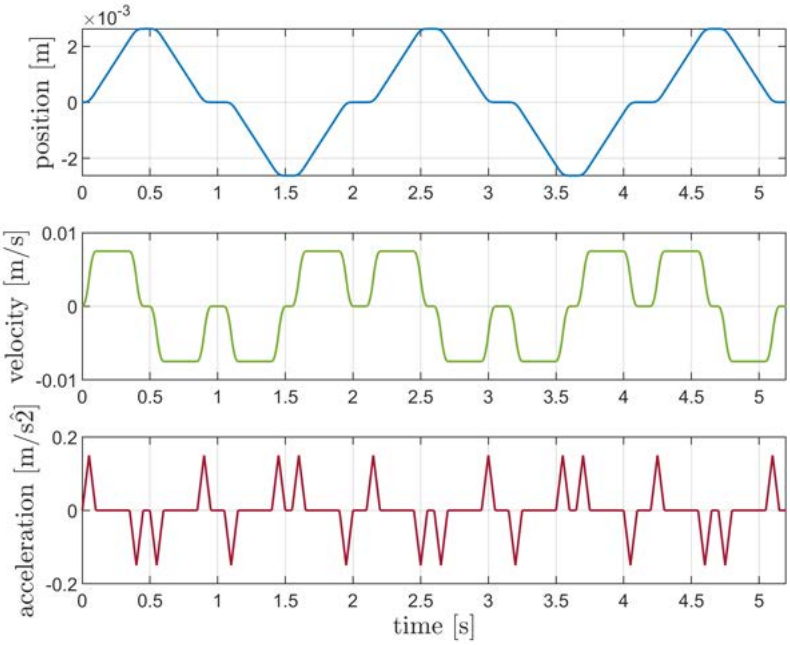


Figure 7.14: New excitation based on cubic trajectory.

that the system is able to follow the reference and the errors of both the plant ([m]) and sensor response ([i]) are in the same magnitude as in the previous (sine) case.

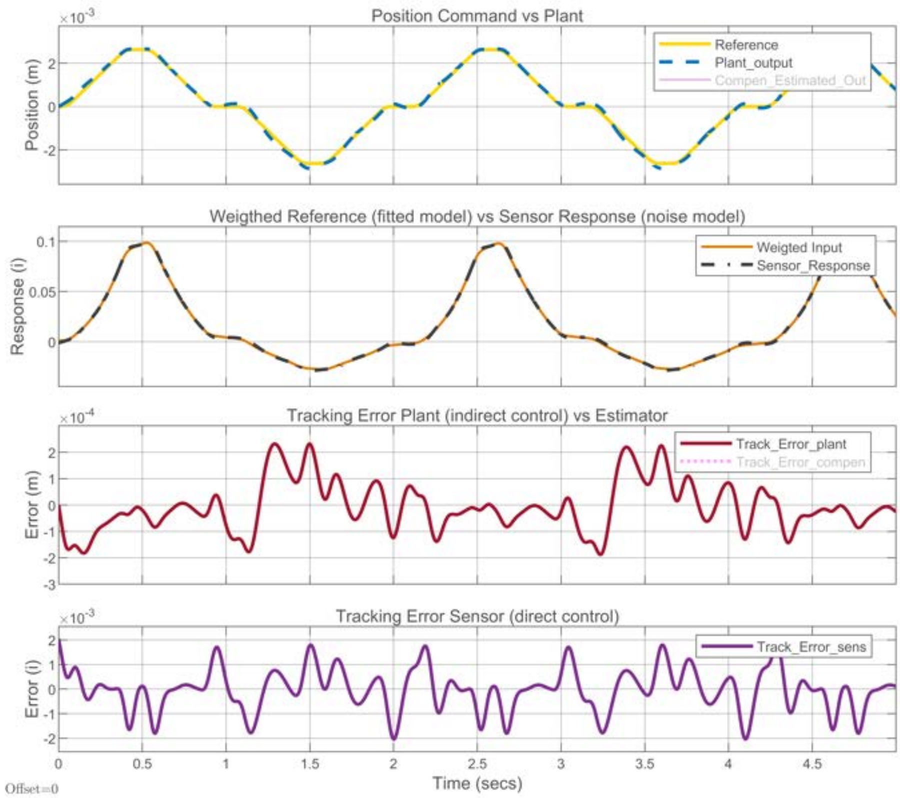


Figure 7.15: Simulation results from 3rd order excitation. From the top (1) Reference position, Plant positon and Estimated plant position respectively. (2) Weighted reference and sensor response, (3) Plant error, and (4) Track error fed into the controller (sensor response).

7.3.2 Output disturbance test

As means to test the impact that tracking the sensor response s or the estimated position $\hat{y}$ has on the plant output in terms of output-disturbance rejection, a series of simulation tests is set were a disturbance in the form of step-function with amplitude of $A_{\text{step}} = [1, 3] \cdot 10^{-3}$ [m] is injected at the time window from $t_{\text{start}} = 1.7$ [s] to $t_{\text{end}} = 2.2$ [s]. The results for both schemes are shown in Fig. 7.16 and Fig. 7.17.

The direct compensation scheme shows lesser performance in terms of plant positioning. Although, from the results, it is shown that both systems are able to recover. The direct compensation scheme, under-performs during the period that the disturbance is active, which is to be expected based on the previous results (Fig. 7.6, 7.7).

In comparison of the overall performance, the tracking of the indirect position control is much better in this scenario. Following the results of sec. 7.2 this is to be expected since the compensator's performance deteriorate significantly in slight changes from the trained excitation. These results, enhance the confidence towards scheme's (b) performance for indirectly controlling the plant position y when the sensor response s is the tracking variable.

Towards a real setup, immediate remedy for the issue of the error offset in the direct compensation, would be to use a *noise* trained model for compensating the non-linearity of the actual 3D printed sensor.

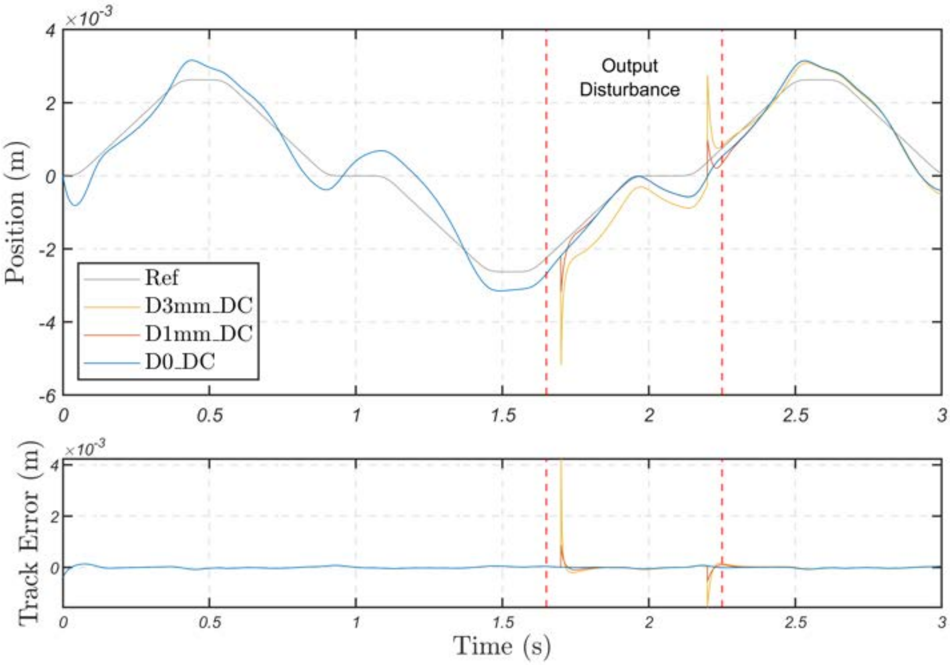


Figure 7.16: Feedback loop with model-based direct compensation for various output disturbances. DC: Direct Compensation (scheme (a)).

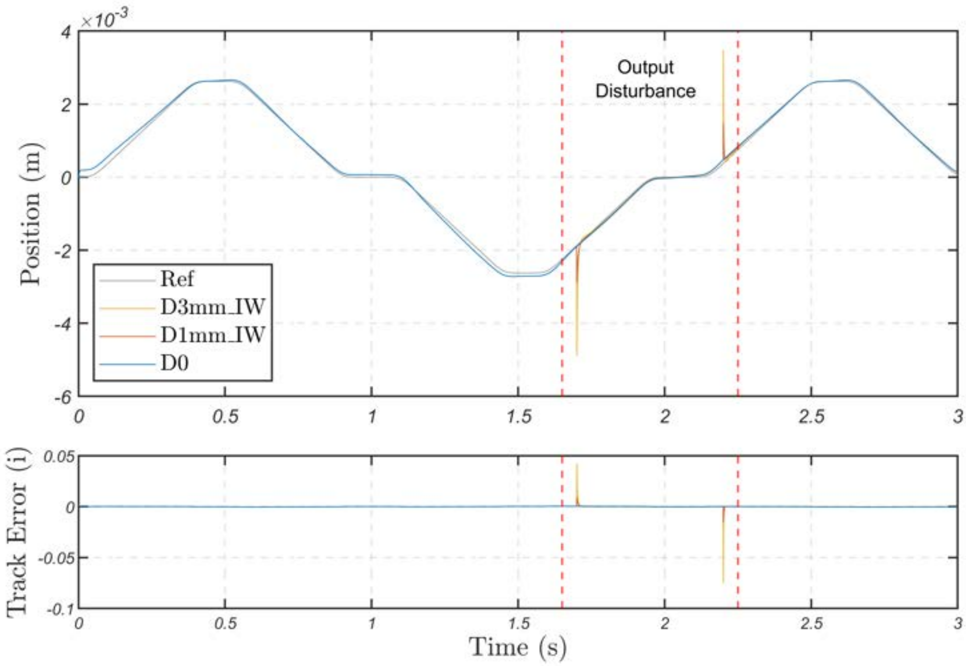


Figure 7.17: Feedback loop with model-based input-weighting compensation for various output disturbances. IW: Input-Weighting compensation (scheme (b)).

7.4 Discussion

As part of discussion, the two schemes are extended with a disturbance-observer estimating filter as means to address system and measurements noise typically present in an actual setup.

With a real application in mind, the results of an ideal simulation, albeit providing an insight, are not expected to entirely, maybe not even partly hold. Typically any real system is subject to external and non-modelled disturbances. It is common then for system and measurement noise to be considered as such disturbances. Of course, shielding a system such that it is able to handle these uncertainties is a subject backed up with decades of literature and practical implementations. Building upon the existing knowledge and using the insight from the ideal simulations, a suggestion can be made towards addressing these disturbances.

One elegant approach towards handling disturbances in rigid systems is proposed in [52]. In this work, a Disturbance-Observer estimating filter is described and a systematic process for deriving said filter is proposed. Practical implementation of this design has shown good results and despite the fact that the system under consideration is not rigid (soft-polymer based sensor), the disturbance rejection process might still be relevant.

For a clear interpretation of the observer process the exposed scheme of Fig. 2 of [52] will be used as a base instead of the feedback interpretation ([52] Fig. 4). Further work on the actual implementation of this suggestion should consider the scheme reduction to obtain the feedback representation since the added value of controller-based stability proofing as well as the intuitive understanding of the controller tuning variables, is a great benefit.

Next, the disturbance-observer scheme can be extended using both of the proposed compensation methods (direct and input-weight).

7.4.1 Direct Compensation with Disturbance-Observer

Fig. 7.18 illustrates the suggested scheme in conjunction with a direct application of the compensator. Similar to the ideal case, the tracking consists of the estimated position $e = r - \hat{y}$. The

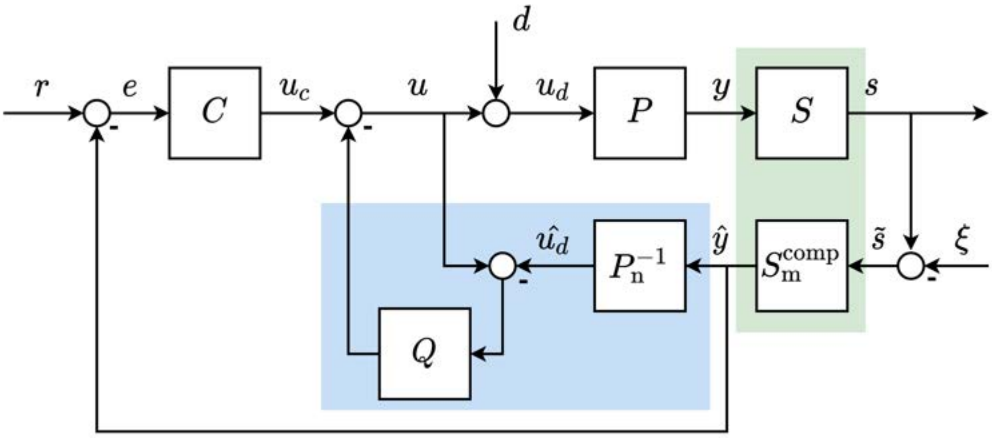


Figure 7.18: Direct compensation extended with disturbance-observer. Compensation (green area), disturbance-observer (blue area).

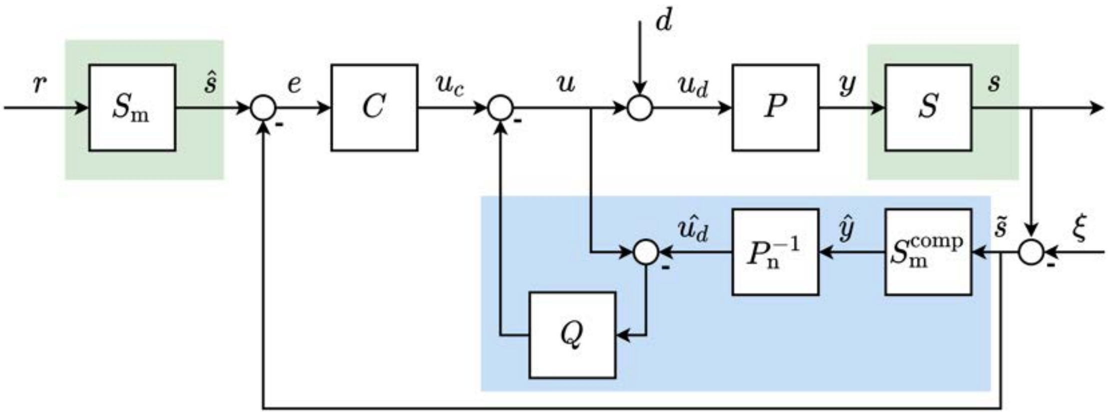


Figure 7.19: Input-Weighting compensation extended with disturbance-observer. Compensation (green area), disturbance-observer (blue area).

observer filter Q acts on the plant input by using the difference between the controller effort and the modelled prediction of the input post-measuring where noise is present. The idea behind the original scheme as seen in [52] aims to eventually cancel the disturbance dynamics and replacing them with the nominal plant P_n.

7.4.2 Input-Weighting Compensation with Disturbance-Observer

Once again, starting with the insight from the ideal simulation, the disturbance rejection extension is shown in Fig. 7.19.

However, in contrast to the ideal case, the important difference in the scheme of 7.19 is that, in spite of the fact that the tracking follows the sensor response and not the estimated position, the inverse compensator is still crucial for deriving the estimated input $\hat{u}$. Meaning, that in either case of the disturbance-observer estimating filter schemes, the issues of the compensator and its limited performance should first be addressed.

Based on the simulation results of schemes (a) and (b), both extensions suggestions look as good candidates for a first approach towards sensor non-linearity compensation in a real setup. Of course, the systematic procedure proposed in [52] for deriving the filter Q should be adapted to new system (with the added process of sensor compensator).

Of course, it is critical to re-state that all the insight gained through simulation results, resides on the assumption that the noise fitted mPL model from [9] sufficiently captures the behavior of the real sensor. The fitting errors present from that model are an inherent *truth* in this ideal scenario (which of course holds for both presented schemes). An actual implementation of these approaches, based on the response of the real sensor, is then necessary before deriving conclusions that will hold in a real-world application.

7.5 Conclusion

The decision is made to assume that the noise trained model of [9] satisfactory captures the real sensor response given different excitations. This model is used as the closest representation of the sensor during simulation. A first simulation is run to log the response of the noise-trained mPL model when the direct output y of the plant P is used for the tracking. These data are then used to fit the first modified mPL model. This model is finally used either to derive a compensator (scheme (a)) or to compensate for the non-linearity by acting on the input r (scheme (b)).

The initial approach of direct application of the compensator within the feedback loop (scheme (a)) is simulated in various tests where the excitation frequency and amplitude deviate in proximity to the ones used for training. It is shown, in spite of the fact that at the trained frequency the compensation partially works (partially linearizing the sensor), that any deviation from these conditions renders the performance inadequate or even unstable (when $A > A_{\text{train}}$).

An alternative concept (scheme (b)) is then proposed where the tracking follows the sensor response instead of the reference and plant position. This is done by weighting the input trajectory with the sensor model. A performance boost is observed in terms of the plant position y following the reference r and evaluated by varying the excitation as well as exciting the system with a cubic trajectory. Attempting to obtain the estimated $\hat{y}$ position with the use of the compensator reveals the possible source of the scheme's (a) failure, where again the compensator properly inverts the non-linear response only at the trained conditions. Although, the response used for feedback in the indirect compensation is a non-linear (hysteretic) signal. Meaning that a PID controller tuned on the linear open-loop would be expected to be sub-optimal, resulting in lower bandwidth and slower controller.

Therefore, a conclusion can be made based on the results of the two proposed schemes that for the direct compensation of scheme (a) the dependence on the inverse model (or invert data trained model) trained from a single excitation severely limits the operation range of the compensation. On the other hand, given a model trained to capture multiple frequencies and amplitudes, the scheme has the prospect to enable often used linear control techniques, assuming the fact that only weak non-linearities remain in the output signal. Towards the indirect compensation of scheme (b), in terms of compensation, the approach is simplified and an inverse model is not required. It is shown that the scheme is less sensitive to excitation deviations and the tracking performance is improved in contrast to the direct compensation, even under the same trained conditions. Despite the latter, due to the non-linear response not being compensated within the feedback loop, the scheme either significantly limits the linearly-tuned PID controller, or requires more elaborate non-linear control techniques.

8 Conclusions

The work towards the investigation of possible control approaches for compensating the hysteretic non-linearity observed in a 3D printed flexible strain-gauge is described. With the presence of hysteresis validated after the examination of the experimental data, the research questions can now be answered as follows:

– *How can the non-linearities of flexible, 3D printed sensors be captured in (a) model(s)?*

In this project, the question is approached by means of model training. Measurement experiments are first carried out. Then, through a parameter estimation process, a model designed to capture said non-linearities is trained. Towards that end, a modified Power-Law model is proposed which, provided with the optimized weights and parameters, is shown to sufficiently capture the hysteretic response.

– *Is it possible to compensate for sensor non-linearities by means of modelling, and, if so, can this compensation be generalized?*

It is shown that by deriving the inverse of the PL model, the resulting compensator eliminates the majority of the hysteretic response at the trained conditions. Towards generalization, since validation over all excitations is practically impossible, in order to obtain a model capable of capturing a wide range of responses, data from an experiment with a random excitation (noise) [9] are shown to generate a mPL model capable of capturing the range of frequencies and amplitudes included in the excitation.

– *What control strategies can be used to successfully exploit the flexible 3D printed sensors?*

With the focus on the 3D printed flexible strain-gauge, two schemes are proposed for controlling the position of a moving-mass system with feedback from the (approximated) 3D printed sensor in a simulated environment. The first scheme applies the compensator directly at the sensor's output (direct compensation) and the estimated plant position (compensator output) is fed into the tracking. The scheme, although partially following the reference, shows poor tracking performance and large sensitivity to slight deviations from the trained conditions. Additionally, a second scheme is proposed that discards the need of an inverse model (or invert data trained forward model) by weighting the input with a forward model. The scheme shows improved performance and less sensitivity to modified excitations. Although, at the price of a lower bandwidth.

8.1 Future Work

The main objective of this work has been to investigate how 3D printed sensors shall be used for control purposes. Following the decision to pursue a model-based compensation, a lot of focus has shifted towards deriving, describing and analyzing the hysteresis model and the compensator. This fact had a clear impact on the work towards control. Therefore, aside of a few suggestions towards the fabrication and modelling, there are still several open questions around the topic of control.

While the sensor fabrication process has been improved, there is room for further improvement as the resistance values of the strain-gauges suggest. This may be done in terms of additional tuning and experimenting with higher/lower resolution nozzles. Moreover, since 3D printing is rapidly evolving and new materials emerge in a rate of every few months, there is high chance that conductive filaments will be further optimized for printing effectively yielding better sensing performance fresh from the spool.

The modified PL models show promising results by sufficiently representing the hysteretic non-linearity, although obtaining a good fit is not a straightforward process. A systematic process of obtaining the initial estimate is needed. Moreover, hysteresis is not the only non-linearity present in the response, creep is expected to be inherited as well due to the material properties. Currently, no focus is put towards validating if the model is still capable (post-modification) of capturing the rate-dependency of polymers. Additionally, experimentation with alternative cost-functions (MSE currently used) may prove fruitful when trying to improve fitting performance.

Towards the compensator, the insight from the control simulations suggest that it is crucial to investigate how to shape the compensator such that it captures a wider range of excitations and not only the one used for its derivation. Concerning the latter, first suggested approach when the actual setup is used (instead of a simulation) is to deploy a noise-trained model as the compensator, as this would be the best candidate for capturing a broader range of the actual sensors response.

With focus on an actual implementation, the development of a process that initializes (synchronizes) the model/compensator with the real sensor might become necessary.

Finally, towards control, since the described schemes are only evaluated during simulation in ideal conditions, the simple schemes of direct and input-based compensation are expected to provide more insight from an actual experiment where the loop is closed with the 3D printed sensor. Furthermore, a proper analysis of the stability of these systems at the operating equilibrium point is required with the optimum goal to derive insight in the global stability of the non-linear system. Moving forward, assuming the compensator issue is addressed, as already proposed, the more sophisticated observer-based designs may be good candidates for addressing system and measurement noise.

3D printed sensors, along with their great prospects, come with great challenges. Addressing the aforementioned open questions, while undoubtedly raising even further research questions, will bring us a step closer to the full potential of embedded 3D printed sensing.

9 Appendix 1

9.1 Hysteresis Compensation of 3D Printed Sensors by a Power Law Model with Reduced Parameters

Hysteresis Compensation of 3D Printed Sensors by a Power Law Model with Reduced Parameters

Dimitrios Kosmas, Martijn Schouten, Gijs Krijnen
Robotics And Mechatronics group, University of Twente, Enschede, The Netherlands
Email: gijs.krijnen@utwente.nl

***Abstract* - We propose a modified Power Law Model [1] for hysteresis compensation. A simplification of the original model, resulting in a lower number of parameters to be estimated, is introduced. It has no nonlinear resistor in the output stage and the nonlinear resistance function in the input section(s) is given by a $\sinh$ function resulting in $3N + 2$ parameters for a model with N input stages. A cantilever beam with two symmetric piezoresistive sensors was 3D printed and shown to exhibit hysteretic behavior. The sensor's differential measurements have been used to obtain training and validation data. We present promising fitting results obtained with a single cell model and 5 parameters only. Finally, the inverse model (compensator) is derived and applied to the experimental data in order to strongly reduce the hysteretic nonlinearity.**

Keywords - 3D-Printing, Hysteresis, Creep, Compensation, Flexible, Soft, Tactile sensor, Power Law, Non-linear

I. Introduction

Rapid growth of Additive Manufacturing (AM) with the possibilities of multi-material printing, along with the recent developments in Soft Robotics has led to the investigation of embedded sensing in soft structures [2]. Fabrication of such sensors has already shown promising results [3], but at the expense of high nonlinearities such as viscoelastic hysteresis [4], [5], hysteretic piezo-resistive response [6] as well as rate-depended phenomena like stress relaxation and creep [5]. For these types of sensors to be incorporated into real-world applications, e.g. for mechatronic purposes, a model that can satisfactorily describe such behavior is needed.

Modeling of hysteretic and rate-dependent phenomena has been the object of an extensive amount of literature since 1935 when F. Preisach introduced his approach of modeling hysteresis in ferromagnetic materials as a sum of infinite rectangular shaped hysteresis operators [7]. Chua & Stromsoe introduced a Lumped Circuit model of hysteresis consisting of non-linear inductors and non-linear resistors [8]. In 1994 Parodi, Storace & Cincotti proposed a piece-wise linear (PWL) ladder circuit model [9] in which the only non-linear elements are the resistive ones, in that way eliminating the extra complexity added by the nonlinear inductors. Biggio & Storace [1] proposed an extended model based on [9], which enables to capture the logarithmic time dependence of the creep relaxation dynamics.

This work was developed within the Wearable Robotics programme, funded by the Dutch Research Council (NWO).

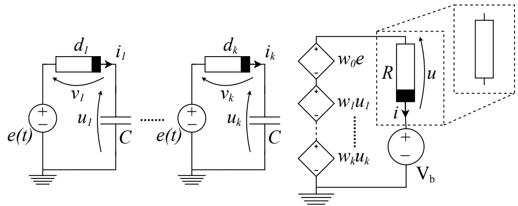


Fig. 1. Power Law model with the proposed circuit modification shown within the dotted area (top right). Adapted from Fig. 1 of reference [1]

In the current work, we attempt to further reduce the complexity of the model of Biggio & Storace, by choosing an alternative way to represent the characteristic of the non-linear resistive elements of the circuit and by changing the fitting procedure. These modifications to the model will first be discussed. Subsequently the model is fitted to measurements obtained on a 3D printed sensor and the inverse of the model is applied to another set of measurement data demonstrating that the model indeed can be used to significantly reduce the hysteresis of a 3D printed sensor.

II. Model

A. Modification

The main background of this model is thoroughly explained in [1] and based on the previous work of [9], [10], [11], [12] (and the references therein). Briefly, the Power Law model attempts to model the hysteretic and creep nonlinearities via an electronic circuit model where the hysteretic memory is modelled by a parallel architecture of $1..N$ sub-circuits (a), fig 1. These sub-circuits contain a non-linear resistor d_k and a capacitor in series with a voltage source with a voltage equal to the input signal $e(t)$. The sub-circuit (b) contains a set of Voltage Controlled Voltage Sources (VCVSs) of which the voltage depends on either a weighted, by w_k, version of the voltage $u_k(t)$ over the capacitors or the input signal $e(t)$ in the first circuit, and a bias voltage V_b. The output signal of the model is the current $i(t)$ going through the non-linear resistor which is in series with the VCVSs. The weights of the VCVS's are estimated through an identification procedure using measurement data. The other elements in the circuit have fixed parameters.

In this work we show that a modified version of this model can be used to reduce the hysteresis and improve the linearity for a sinusoidal input signal using only a single instance of the sub-circuit a. This modified version of the model discards the second non-linear element by fixing it to a value of $R = 1$ (panel (b)). Also it assumes that the non-linear resistor (panel (a)) can be expressed as a weighted hyperbolic sine function

$$i_k = f(u_k) = I_\mathrm{S}\left(e^{\frac{u_k}{V_\mathrm{T}}} - e^{\frac{-u_k}{V_\mathrm{T}}}\right), \quad k = (1, 2, ..., N) \tag{1}$$

Finally we optimise not only the weights w_k of the VCCSs, but also the shape of the non-linear resistor by optimizing the parameters V_T and I_S. This optimisation is done using a non-linear optimization procedure that finds the minimum of a constrained nonlinear multivariable function [13].

The model can be represented as a non-linear state-space equation

$$\begin{aligned} \dot{u}_k &= \frac{1}{C} f(v_k) = \frac{1}{C} f(e - u_k) \\ iR &= w_0 e + \sum_{k=1}^{N} w_k u_k - V_b \end{aligned} \tag{2}$$

Hence, the inverse (compensator) output can be represented as

$$\begin{aligned} \dot{u}_k &= \frac{1}{C} f\left(\frac{iR}{w_0} - \sum_{j=1}^{N} \frac{w_j}{w_0} u_j + \frac{V_\mathrm{b}}{w_0} - u_k\right) \\ e &= \frac{iR}{w_0} - \sum_{j=1}^{N} \frac{w_j}{w_0} u_j + \frac{V_\mathrm{b}}{w_0} \end{aligned} \tag{3}$$

B. Identification

In order to estimate the remaining weights ($w_0, w_1, V_\mathrm{b}, I_\mathrm{S}$, and V_T), a set of measurement data is obtained for $\hat{e}(t)$ and $\hat{i}(t)$ (see section III). The number of cells N is to be provided. A cost function is defined as the summed squared error

$$SSE = \sum_{t=0}^{t_\mathrm{end}} \left(\hat{i}(t) - i(\hat{e}(t))\right)^2 \tag{4}$$

where $\hat{i}(t)$ is the measured, and $i(\hat{e}(t))$ the output modelled for the input $\hat{e}(t)$.

For the minimization, MATLAB's `fmincon` is used in order to estimate the weights that minimise the cost function defined in 4. An interior-point based algorithm [13] is employed, limited such that the sum of all of the weights is smaller than 10^2.

C. Verification

As a measure for the amount of hysteresis of a sensor, the average area encircled in a plot of the normalised input versus the normalised input, will be used. To calculate this area the following equation is used.

$$\begin{aligned} \hat{A} &= \frac{1}{n} \int_{\hat{x}_{t=0}}^{\hat{x}_{t=n \cdot T}} \hat{y} \cdot d\hat{x} \\ &\text{with:} \\ \hat{x} &= \frac{x - \min(x)}{\max(x) - \min(x)} \\ \hat{y} &= \frac{y - \min(y)}{\max(y) - \min(y)} \end{aligned} \tag{5}$$

Where x is the input displacement of the sensor, y is the output signal of the sensor, T is the period of the excitation signal and n is an integer representing the number of periods in the test signal.

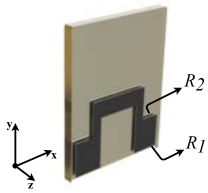


Fig. 2. 3-D CAD render of the differential sensor exposing top & bottom resistive traces (co-ordinate frame representing the printing orientation).

Fig. 3. Fabricated sensor on top of a PVA layer.

III. Measurements

A. Sensor

The sensor fabrication was largely based on the design of the previous work of [3], where it was demonstrated that differential measurements can improve the linearity of the sensor. Here we follow the same concept by implementing two symmetric piezo-resistive elements on both the top and bottom faces of the sensor (fig. 2). The piezo resistors are based on a carbon black filled thermoplastic polyurtherane (TPU) by Ninjatek called EEL [14], as shown in figure 3.

In contrast to the previous sensor, the sensor used in the current work was printed on a raft of PVA. This was done because the first layer often has different properties than the succeeding layers, which causes asymmetry in the sensor.

The measured resistances of the piezo-resistive traces, after allowing a settling period, were found to be $22.4\,\mathrm{k\Omega}$ (top) and $15.7\,\mathrm{k\Omega}$ (bottom).

B. Set-up

The sensor is mechanically loaded using a linear actuator (SMAC LCA25-050-15F) running a position control loop generating a sine with a frequency of $0.5\,\mathrm{Hz}$ and an amplitude of $3\,\mathrm{mm}$. To read out the sensor, it is placed in a half-bridge

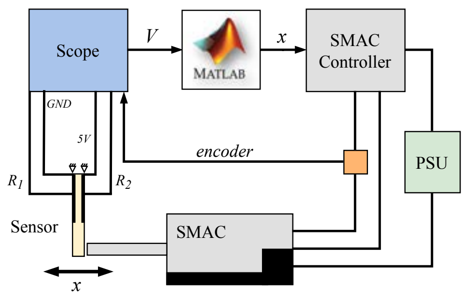


Fig. 4. Overview of the experiment set-up (orange box represents the *splitter* cable for the extraction of encoder's channels A & B signals).

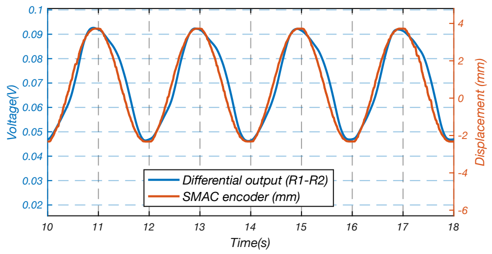


Fig. 5. Differential measurement(left y-axis) lags the commanded commanded position (right y-axis).

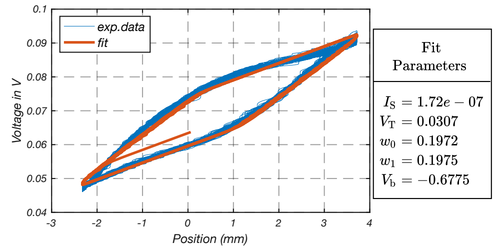


Fig. 6. Experimental data and Modified PL model fit.

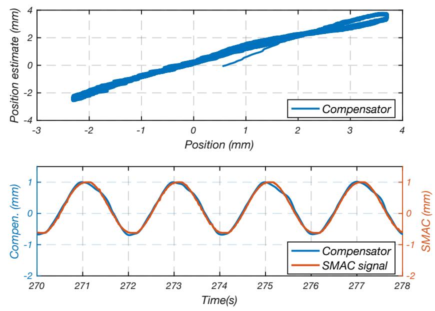


Fig. 7. Inverse model output versus measured position and response of the compensator.

configuration using a $10\,\mathrm{k\Omega}$ resistor and a $2\,\mathrm{V}$ DC voltage source. The half-bridge output voltage is measured using a Picoscope 5443B running at a sample frequency of $2\,\mathrm{MHz}$. In order to be able to guarantee a perfect time synchronization of the half bridge measurement with the position readout, the encoder signal from the SMAC actuator controller is extracted and read by the PicoScope.

At the sensor channels, a down-sampling ratio of 5 along with a digital zero-phase low-pass 2^{nd} order Butterworth filter is applied with a cut-off frequency of $f_c = 20\,\mathrm{Hz}$. Due to an observed high-frequency interference of the SMAC controller, a hardware RC low-pass filter is added with $R = 10\,\mathrm{k\Omega}$ and $C = 15\,\mathrm{nF}$. An overview of the setup is presented in fig 4.

IV. Results

Two data sets of $\hat{e}(t)$ (SMAC position) and $\hat{\imath}(t)$ (half bridge output voltage) were obtained by two distinct experiments of $5\,\mathrm{min}$ duration each. Cross-validation is used, with one set of data for fitting and one for validation respectively. The differential measurement versus the input position, see figure 5, is forming the hysteresis loop. Using the measured time series $\hat{e}(t)$ and $\hat{\imath}(t)$ the model is nonlinearly fitted with the first set. For the computation, sub-circuit (a) is comprised with a single cell ($N = 1$). The capacitance is fixed at $C = 1\,\mathrm{F}$. The initial conditions are given as: $I_S = 0.1$, $V_T = 0.2$, $w_0 = 1$, $w_1 = 0.4$ and $V_b = -0.6$. Hence, a total of five parameters are to be optimized through the optimization process. The model fit is shown in fig 6.

The inverse model, given the optimized parameters, is applied to the second set of data and the output is plotted against the experiment data in fig 7. The SINAD and the normalised hysteresis when compared to the SMAC encoder signal as defined in equation 5 are calculated for the most important signals (Table I).

V. Discussion and Conclusions

The modified model shows a good fit against the experimental data, correctly representing the hysteretic nonlinearity. MATLAB's `fmincon` proves that a good fit is possible. However a more efficient, in terms of computation time, approach is likely required before the model can be adapted in an online estimation.

Furthermore, the inverse model is applied and is shown to provide a good estimation of the original position (fig. 7-top). Subsequently, the response of the compensator is shown to reduce the hysteretic non-linearity (fig. 7-bottom-) as compared to fig. 5. As far as the linearity is concerned, only a slight improvement was observed. (table I).

In this experiment, an empirical model is constructed for a system that consists of both a cantilever beam fabricated out of a viscoelastic material as well as a piezo-resistive sensing element. In order to be able to model these elements separately the force applied on the cantilever should be measured as well.

Further research is required in order to validate the simplified model's performance. The presented results hold for a specific frequency and amplitude. For different cases the need to employ a larger number of N cells might be necessary. The ability to represent inner loops and the model's performance for excitation at different frequencies and amplitudes is still to be investigated. Preliminary results are promising, albeit at the expense of requiring multiple cells instead of one.

TABLE I. SINAD

Signal	SINAD (dB)	$\hat{A}$ (%)
SMAC encoder signal	32.2	0
R_1	4.9	-10
R_2	5.7	15
$R_1 - R_2$	18.4	-18
Fit	23.6	-20
Compensated	20.6	0.8

References

[1] M. Biggio, A. Oliveri, F. Stellino, M. Parodi, and M. Storace, "A circuit model of hysteresis and creep," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 5, pp. 501–505, 2015.

[2] A. Dijkshoorn, P. Werkman, M. Welleweerd, G. Wolterink, B. Eijking, J. Delamare, R. Sanders, and G. J. Krijnen, "Embedded sensing: Integrating sensors in 3-D printed structures," *Journal of Sensors and Sensor Systems*, vol. 7, no. 1, pp. 169–181, 2018.

[3] M. Schouten, B. Prakken, R. Sanders, and G. Krijnen, "Linearisation of a 3d printed flexible tactile sensor based on piezoresistive sensing," in *2019 IEEE SENSORS*, Oct 2019, pp. 1–4.

[4] H. Banks, "A brief review of some approaches to hysteresis in viscoelastic polymers," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 69, no. 3, pp. 807–815, 2008.

[5] H. Qi and M. Boyce, "Stress-strain behavior of thermoplastic polyurethanes," *Mechanics of Materials*, vol. 37, no. 8, pp. 817–839, 2005.

[6] J. Shintake, E. Piskarev, S. H. Jeong, and D. Floreano, "Ultrastretchable strain sensors using carbon black-filled elastomer composites and comparison of capacitive versus resistive sensors," *Advanced Materials Technologies*, vol. 3, no. 3, p. 1700284, 2017.

[7] F. Preisach, "Uber die magnetische nachwirkung," *Zeitschrift fur Physik*, vol. 94, no. 5-6, pp. 277–302, 1935.

[8] L. Chua and K. Stromsmoe, "Lumped-circuit models for nonlinear inductors exhibiting hysteresis loops," *IEEE Transactions on Circuit Theory*, vol. 17, no. 4, pp. 564–574, 1970.

[9] M. Parodi, M. Storace, and S. Cincotti, "A pwl ladder circuit which exhibits hysteresis," *International Journal of Circuit Theory and Applications*, vol. 22, no. 6, pp. 513–526, 1994.

[10] S. Bobbio and G. Marrucci, "A possible alternative to preisach's model of static hysteresis," *Il Nuovo Cimento D*, vol. 15, no. 5, pp. 723–733, 1993.

[11] M. Storace and M. Parodi, "On the representation of static hysteresis curves by a pwl ladder circuit," *International Journal of Circuit Theory and Applications*, vol. 26, no. 2, pp. 167–177, 1998.

[12] A. Oliveri, M. Lodi, F. Stellino, and M. Storace, "Modeling and compensation of hysteresis and creep: The hystool toolbox," *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018.

[13] "Matlab optimization toolbox," Version 8.4 (R2019b), mathWorks, Natick, MA, USA.

[14] Fenner Drives, "Eel 3d printing filament," Report, 2018. [Online]. Available: http://ninjatek.com/wp-content/uploads/2018/12/Eel-TDS.pdf

9.2 Hysteresis Compensation of 3D Printed Sensors Using a Power Law Model for Various Input Signals

Hysteresis Compensation of 3D Printed Sensors Using a Power Law Model for Various Input Signals

Martijn Schouten, Dimitrios Kosmas, Gijs Krijnen
Robotics And Mechatronics group, University of Twente, Enschede, The Netherlands
Email: m.schouten@utwente.nl

***Abstract* - We calculated the stability criterion for the modified power law model and subsequently adjust the model such that it's stability can be guaranteed. We applied both a sinusoid, a chirp, an exponentially decaying sine and bandwidth limited noise position excitation to a 3D printed symmetric piezoresistive cantilever and measure the differential response. A modified power law model is fitted directly to the inverse of the sensor data, to directly obtain a compensator for the sensor. The result was a stable compensator that reduced the hysteresis of the 3D printed sensor.**

Keywords - 3D-Printing, Hysteresis, Compensation, Flexible, Soft, Tactile sensor, Power Law, Non-linear, Stability, Recurrent, Shallow, Neural Network

I. Introduction

3D printing is a promising technique for the fabrication of sensors, since the technique allows sensors to be produced cost effectively in low volumes, at high integration levels and considerable complexity [1], [2]. The technique is also well suited for the fabrication of soft sensors [3], [4].

A common material used in 3D printing of soft sensors is TPU, which is known to show visco-elastic behaviour [5]. Furthermore the piezoresistive elements that are often used, show a non-linear relationship and hysteresis [6]. Together both effects often result in hysteretic sensing behaviour.

To model hysteretic behaviour a circuit model was proposed by Biggio et al [7]. A simplified version of this model was shown to be able to significantly reduce the hysteresis in a 3D printed cantilever beam for a sinusoidal input [8]. In this paper the stability of this model is further improved and it is shown that this model can also be used to reduce the hysteresis for more complicated excitation signals.

II. Model

The model that is used is based on the model described in [7]. The model has an electrical circuit representation which is shown in figure 1. The model consists of multiple cells consisting of a non-linear resistor and a capacitor (with C=1 F). The voltage on the capacitors in subcircuit (a) is linked to the voltage of the voltage controlled voltage sources (VCCS's) in sub-circuit (b). These VCCS's, a VCCS directly linked to the

This work was developed within the Wearable Robotics programme, funded by the Dutch Research Council (NWO).

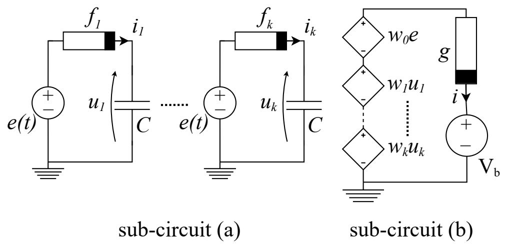


Fig. 1. A circuit representation of the model. Image courtesy of [7]

input and a fixed voltage source determine the voltage over a second non-linear resistor. The current through this second non-linear resistor is the output of the model. This circuit can be represented using a non-linear state space equation [8]:

$$\begin{aligned} \dot{u}_k &= f(e - u_k) \\ i &= g(w_0 e + \sum_{k=1}^{N} w_k u_k - V_b) \end{aligned} \tag{1}$$

where e is the input signal, i is the output, u_k are the voltages on the capacitors and f and g are non-linear functions. w_k and V_b are weights of the model and are to be fitted together with the parameters of the non-linear functions.

A. Linearisation

This non-linear state space equation can be linearised, around a certain state $u_{k,0}$ and input e_0, by using a first order Taylor series approximation of the non-linear functions.

$$\begin{aligned} \dot{u}_k &= f_{k,0} + f'_{k,0} e - f'_{k,0} u_k \\ i &= g_0 + g'_0 w_0 e + g'_0 \sum_{k=1}^{N} w_k u_k - g'_0 V_b \end{aligned} \tag{2}$$

With the following definitions:

$$\begin{aligned} f_{k,0} &= f_k(u_{k,0} - e_0), \qquad g_0 = g(w_0 e_0 + \sum_{k=1}^{N} w_k u_{k,0} - V_b) \\ f'_{k,0} &= f'_k(u_{k,0} - e_0), \qquad g'_0 = g'(w_0 e_0 + \sum_{k=1}^{N} w_k u_{k,0} - V_b) \end{aligned} \tag{3}$$

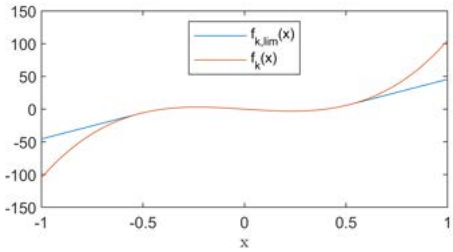


Fig. 2. Graphical illustration of the limit applied by equation 9

This can be written as a linear state space representation.

$$\begin{aligned}\dot{x} &= Ax + Be\\ i &= Cx + De\end{aligned} \tag{4}$$

With:

$$A = \begin{bmatrix} -f'_{1,0} & 0 & \dots & 0 & f_{1,0} & 0\\ 0 & -f'_{2,0} & \dots & 0 & f_{2,0} & 0\\ \vdots & \vdots & \ddots & & & \\ 0 & 0 & & -f'_{N,0} & f_{N,0} & 0\\ 0 & 0 & & 0 & 0 & 0\\ 0 & 0 & & 0 & 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} f'_{1,0}\\ f'_{2,0}\\ \vdots\\ f'_{N,0}\\ 0\\ 0\end{bmatrix},\ x = \begin{bmatrix} u_1\\ u_2\\ \vdots\\ u_N\\ 1\\ 1\end{bmatrix} \tag{5}$$

$$C = [g'_0 w_1 \quad g'_0 w_2 \quad \dots g'_0 w_N \quad 0 \quad g'_0 V_b + g_0],\ D = g'_0 w_0$$

B. Discretization

This continuous state space equation can be discretized into the form:

$$\begin{aligned}x(k+1) &= A_d x(k) + B_d e(k)\\ i &= C_d x(k) + D_d e(k)\end{aligned} \tag{6}$$

When each matrix exponential is approximated using only the first two terms, only the A matrix will change during discretization [9, p. 315]. This A matrix will change in the following way

$$A_d = I + A\Delta t \tag{7}$$

C. Stability

This system is asymptotically stable in case the magnitude of all eigenvalues of A_d are less than unity [9, p. 329]. The eigenvalues of A_d are located at $\lambda_k = 1 - f'_{k,0}\Delta t$ and at zero. This means the system is asymptotically stable in case:

$$0 < f'_{k,0} < \frac{2}{\Delta t} \tag{8}$$

To prevent instability the function f_k is limited such that:

$$f_{k,\text{lim}} = \begin{cases} f_k(x^+) + \frac{2}{\Delta t}(x - x^+) & \text{for } x > x^+\\ f_k(x) & \text{for } x^- > x > x_+\\ f_k(x^-) + \frac{2}{\Delta t}(x - x^+) & \text{for } x < x^- \end{cases} \tag{9}$$

where x^- and x^+ are respectively the minimum and the maximum value of the set of solutions of the equation $f'(x) = \frac{2}{\Delta t}$. Figure 2 graphically shows the implication of this limit. Note that this limitation allows the model to be unstable for $x^- > x > x_+$, but that such an instability always will cause the model to enter a part of the function where it is stable again, since the model is stable for both $x \to \infty$ and $x \to -\infty$.

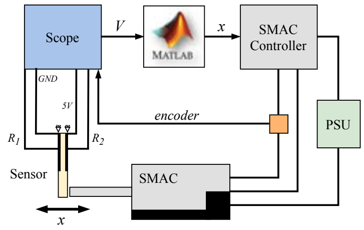


Fig. 3. Graphical illustration of the used measurements setup [8] (©[2020] IEEE).

III. Measurements

The used sensor and setup were previously described in [8] and [10]. In summary a linear actuator (SMAC LCA25-050-15F) is used to deform a 3D printed cantilever and a differential measurement of the resistance of piezoresistive elements on both sides of the cantilever is used to measure this deformation, see figure 3. An optical encoder is used as reference of the applied position signal. In this work two coupled Handyscope HS5 scopes sampling at $6.25\,\text{MHz}$ were used to measure the encoder signal and the voltage of two half bridges both with a $560\,\Omega$ resistor, one of the piezoresistive elements and a DC supply voltage of $1\,\text{V}$. A high sampling frequency was used in order to be able to have enough bandwidth to measure all the pulses of the $5\,\mu\text{m}$ resolution encoder.

Each measurement was executed twice in order to obtain one data set to fit the model to and one to validate the model on. A 2nd order zero-phase $20\,\text{Hz}$ Butterworth filter was applied to the measured voltages and the decomposed encoder signal in order to reduce the noise in the measurement. After applying this filter all signals are downsampled to $40\,\text{Hz}$ to speed up the model execution.

The difference between the relative resistance changes of both piezoresistors is calculated from the measured half-bridge voltages and used as the sensor output. The mean of the data was subtracted from both the sensor output and the encoder signal, to reduce the influence of drift that occurred in between the measurements. To normalise the data, the fitting and validation data are divided by the maximum of the absolute value of the fitting signals. The used excitation signals were:

1) A $0.5\,\text{Hz}$ sinusoid of $200\,\text{s}$ duration with an amplitude of $3\,\text{mm}$.
2) A Chirp signal with a frequency exponentially increasing over a period of $100\,\text{s}$ from $0.1\,\text{Hz}$ to $3\,\text{Hz}$ and an amplitude of $3\,\text{mm}$
3) A sinusoid with a frequency of $0.5\,\text{Hz}$ and an amplitude that exponentially decreases over $200\,\text{s}$ from $2\,\text{mm}$ to $100\,\mu\text{m}$.
4) Gaussian noise filtered with a 500^{th} order FIR low-pass filter with a $-3\,\text{dB}$ cut-off frequency of $4\,\text{Hz}$

In the rest of this paper these signals will be abbreviated as Sine, Chirp, Decay and Noise respectively. The sensor output is used as input for the model and the parameters of the model are optimised such that the model output predicts the position of the encoder. In this way the model can be used directly as a compensator for the hysteresis and non-linearity of the data.

TABLE I. LEFT: INITIAL CONDITIONS OF THE USED POLYNOMIAL FUNCTIONS. RIGHT: ORDINARY COEFFICIENTS OF DETERMINATION

	0	1	2	3
$f(x)$	-1	0.1	0.1	1
$g(x)$	0.1	1	0.1	1

	Model	Linear
Sine	0.990	0.928
Chirp	0.984	0.945
Decay	0.987	0.710
Noise	0.958	0.882

TABLE II. FITTED WEIGHTS

	Sine	Chirp	Decay	Noise
f_0	-2.516	-0.469	-0.258	-4.719
f_1	-2.666	-1.712	0.663	3.412
f_2	1.975	-2.228	-0.002	-1.244
f_3	1.953	3.125	0.008	1.469
g_0	0.100	-0.369	-0.779	-0.119
g_1	1.195	1.094	1.320	1.125
g_2	0.006	0.006	-0.002	0.006
g_3	0.475	-0.384	-0.240	-0.259
w_0	1.000	1.516	0.984	1.156
w_1	-0.492	-0.531	-0.875	-0.406
V_b	0.135	-0.537	-0.302	0.010

The fit is done using MATLAB's `patternsearch` running on all 6 cores of an Intel I7 9850 for 60 s. Executing the model with the sine data takes 18 ms. For both $f(x)$ and $g(x)$ a 3$^{\text{th}}$ order polynomial of the following form is used:

$$f(x) = f_0 + f_1 x + f_2^2 x^2 + f_3^3 x^3 \tag{10}$$

The initial conditions of f and g can be found in table I on the left. In all cases $N = 1$ and the initial condition of the remaining weights w_0, w_1 and V_b were 1, -1 and 0.01 respectively. As a reference of the performance, the model is compared to a linear fit, using MATLAB's `polyfit`.

IV. RESULTS

Figure 4 shows the hysteresis in the measured signals. Figure 5 shows the hysteresis of the model and the linear fit after they are applied to the data. The ordinary coefficient of determination between the encoder position and the position predicted by the the model as well as a linear fit can be found in table I on the right.

The fitted model parameters can be found in table II. The shape of the corresponding polynomials $f_{\text{lim}}(x)$ and $g(x)$ can be found in figure 6. Without the limits set in equation 9 the fitted model of the chirp is unstable with the used validation data-set.

V. DISCUSSION AND CONCLUSIONS

A stability criterion for a modified power law model was derived. The model was subsequently adjusted to meet this criterion and it was shown that the model reduces the hysteresis of a 3D printed sensor in comparison to a linear fit, for several input signals. The used model only has 11 parameters and although the training was computationally intensive, the execution of the model was not. This research only shows a reduction of the hysteresis for the same sensor under the same conditions as was used during training. Further research is needed to determine the applicability in other situations.

Although a differential measurement is employed, which is expected to remove drift common to both strain gauges, it was necessary for a good fit that the mean of the signal was subtracted from both the fitting and the validation signals. Since the model itself does not remove inaccuracies due to drift further research on reducing drift is required when the sensors should be used for absolute force measurements.

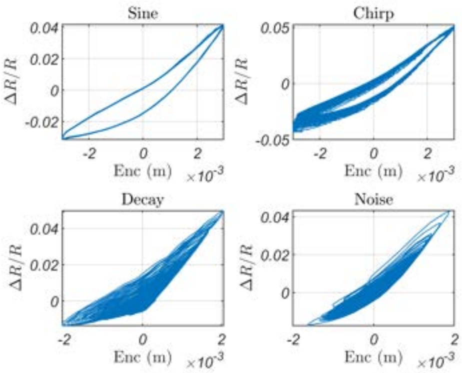


Fig. 4. Difference between the relative change in resistance of the two piezoresistive elements ($\Delta R/R$) against the position measured by the encoder (Enc) for different excitation signals

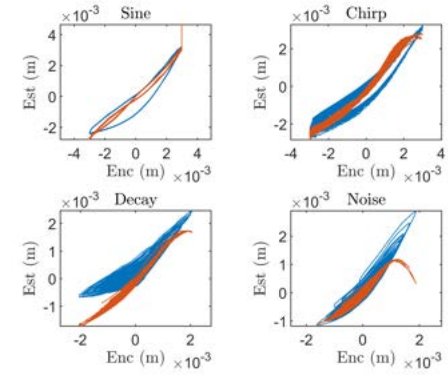


Fig. 5. Position estimated by the model (red) and a linear fit (blue) versus the position measured by the encoder (Enc) for different excitation signals

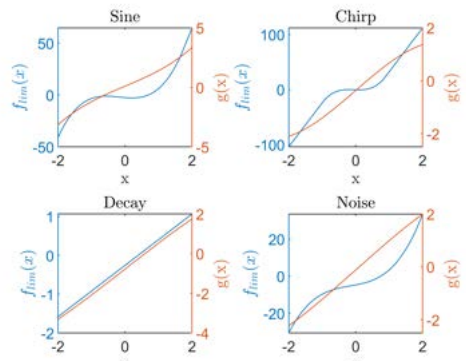


Fig. 6. The shape of the fitted f_{lim} and g functions

REFERENCES

[1] T. Han, S. Kundu, A. Nag, and Y. Xu, "3D printed sensors for biomedical applications: A review," *Sensors (Switzerland)*, vol. 19, no. 7, 2019.

[2] A. Dijkshoorn, P. Werkman, M. Welleweerd, G. Wolterink, B. Eijking, J. Delamare, R. Sanders, and G. J. Krijnen, "Embedded sensing: Integrating sensors in 3-D printed structures," *Journal of Sensors and Sensor Systems*, vol. 7, no. 1, pp. 169–181, 2018.

[3] G. Wolterink, R. Sanders, and G. Krijnen, "A flexible, three material, 3D-printed, shear force sensor for use on finger tips," *Proceedings of IEEE Sensors*, vol. 2019-Octob, pp. 2–5, 2019.

[4] M. O. F. Emon, F. Alkadi, D. G. Philip, D. H. Kim, K. C. Lee, and J. W. Choi, "Multi-material 3D printing of a soft pressure sensor," *Additive Manufacturing*, vol. 28, no. May, pp. 629–638, 2019. [Online]. Available: https://doi.org/10.1016/j.addma.2019.06.001

[5] H. J. Qi and M. C. Boyce, "Stress-strain behavior of thermoplastic polyurethanes," *Mechanics of Materials*, vol. 37, no. 8, pp. 817–839, 2005.

[6] J. Shintake, E. Piskarev, S. H. Jeong, and D. Floreano, "Ultrastretchable Strain Sensors Using Carbon Black-Filled Elastomer Composites and Comparison of Capacitive Versus Resistive Sensors," *Advanced Materials Technologies*, vol. 3, no. 3, pp. 1–8, 2018.

[7] M. Biggio, A. Oliveri, F. Stellino, M. Parodi, and M. Storace, "A circuit model of hysteresis and creep," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 5, pp. 501–505, 2015.

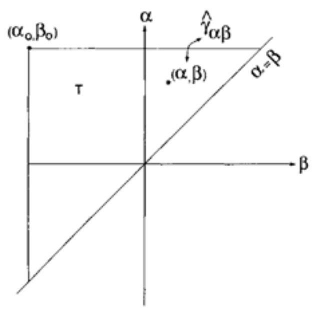
[8] D. Kosmas, M. Schouten, and G. Krijnen, "Hysteresis Compensation of 3D Printed Sensors by a Power Law Model with Reduced Parameters," *FLEPS 2020 - IEEE International Conference on Flexible and Printable Sensors and Systems, Proceedings*, 2020.

[9] K. Ogata, *Discrete-Time Control Systems (2nd Ed.)*. USA: Prentice-Hall, Inc., 1995.

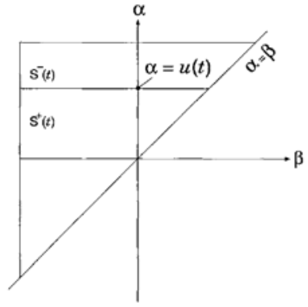
[10] M. Schouten, B. Prakken, R. Sanders, and G. Krijnen, "Linearisation of a 3d printed flexible tactile sensor based on piezoresistive sensing," in *2019 IEEE SENSORS*, Oct 2019, pp. 1–4.

9.3 Preisach basic principle

In order to properly describe the Preisach model some definitions need to be mentioned. The mathematical analysis of the Preisach model is based on its geometric interpretation. This interpretation is based on the fact that an one-to-one relationship exists between the $\hat{\gamma_{\alpha\beta}}$ operators and the points (α,β) of the half-plane $\alpha \geq \beta$, as show in Fig. 9.1a. This would mean that each point in the $\alpha \geq \beta$ plane, is decribed with a single $\hat{\gamma_{\alpha\beta}}$ operator, whose "up" and "down" switching values are the α and β coordinates of the point (Fig. 2.2).



(a) Preisach $\alpha\beta$ plane of the switching points.



(b) $\alpha\beta$ plane S sub-sets.

Figure 9.1: Preisach $\alpha\beta$ plane.

At each moment in time the triangle T (formed by the hypotenus $\alpha = \beta$) is split into two sets (S^- and S^+) by the line formed from current input ($a = u(t)$). Assuming a sequence of monotically increasing input values until u_{max} and of monotically decreasing input values until u_{min}, the staircase interface L is formed as shown in Fig. 9.3a. The conclusion that can be drawn is that at each instant of time the triangle T is subdivided in the two sets, the S^- set for which the points (α,β) correspond to $\hat{\gamma_{\alpha\beta}}$ operators in the "down" position, and the S^+ set for which the points (α,β) correspond to $\hat{\gamma_{\alpha\beta}}$ operators in the "up" position. Therfore at each t the Preisach integral from eqn. 2.1 can be represented as:

$$f(t) = \hat{\Gamma}\mu(t) = \int\int_{S^+(t)} \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}u(t)\,d\alpha\,d\beta + \int\int_{S^-(t)} \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}u(t)\,d\alpha\,d\beta \tag{9.1}$$

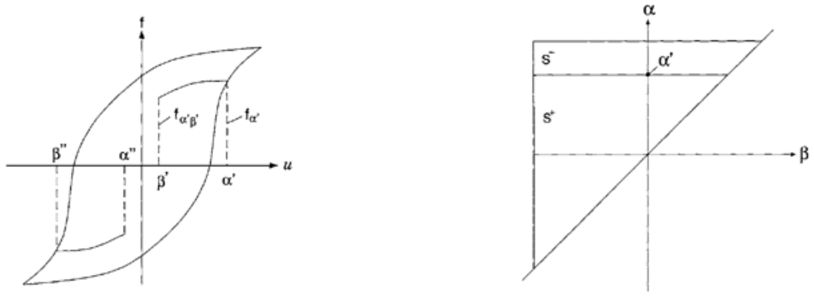
Since $\hat{\gamma_{\alpha\beta}} = 1$ for $(\alpha,\beta) \in S^+(t)$ and $\hat{\gamma_{\alpha\beta}} = -1$ for $(\alpha,\beta) \in S^-(t)$, it follows:

$$f(t) = \int\int_{S^+(t)} \mu(\alpha,\beta)\,d\alpha\,d\beta - \int\int_{S^-(t)} \mu(\alpha,\beta)\,d\alpha\,d\beta \tag{9.2}$$

It is shown then that the resulting shape of the straicase interface L depends on the past extremum values of the input because the vertices of these inputs are shaping the interface. The instantaneous value of the output in this case, depends on the subdivision of the limiting triangle T in the two subsets S^- and S^+. This is in fact the mechanism in the Preisach model of how the past input extrema leave their mark upon the future [17].

9.4 Preisach transition curves

The goal of the identification problem for the Preisach model is the derivation of the distribution function $\mu(\alpha,\beta)$. Towards that end a set of first-order transition curves were used. In order to provide a definition on what these curves represent, lets first assume that the input $u(t)$ is decreased to value less than β_0. This will bring the hysteresis nonlinearity to a point of negative saturation. Then, the input is monotically increased to a point where $u(t) = \alpha'$ (see Fig. 9.2b). While the input increases the asceding branch of the major loop is tracked (Fig. 9.2a), also called the limited ascending branch since no branch exist below it.



(a) Input-output generalized Preisach hysteresis as shown in [17].

(b) $\alpha\beta$ plane first state of monotically increasing input.

Figure 9.2: Preisach transition curves.

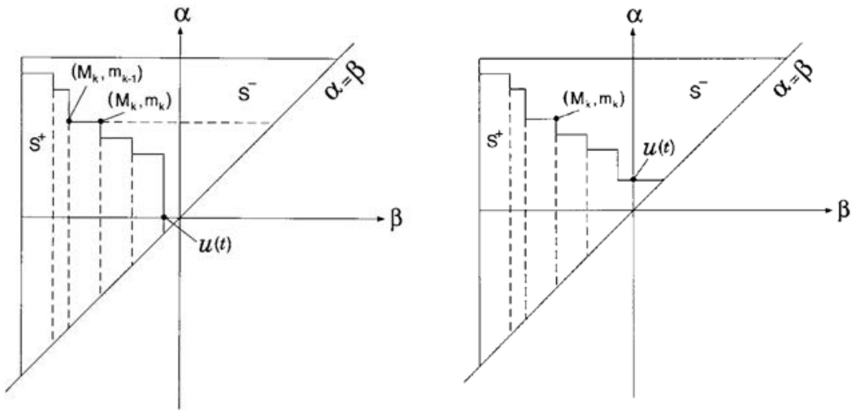
The so-called first-order transition curve is generated when the aformentioned monotic increase of the input is succeded by a monotonic decrease. The term *"first-order"* stems from the fact that each of these curves is formed after the reversal of the input.

9.5 Preisach digitization

The digitized equations are shown in 9.3. Here, f_1 represents the output for a monotonically increasing input, and f_2 the output for a monotonically decreasing input. These explicit expressions are expressed in terms of experimental measured data. An extensive explanation for the derivation of the relation between the following equations and the experimental data is shown in the *"Identification Problem for the Preisach model. Representation Theorem."* section of the Chapter 1 of Mayergoyz's book.

$$
\begin{aligned}
f_1(t) &= -f^+ + \sum_{k=1}^{n-1} (f_{M_k m_k} - f_{M_k m_{k-1}}) + f_{M_n u(t)} - f_{M_n m_{n-1}}, \\
f_2(t) &= -f^+ + \sum_{k=1}^{n-1} (f_{M_k m_k} - f_{M_k m_{k-1}}) + f_{-m_{n-1}} - f_{-m_{n-1}, -u(t)}
\end{aligned}
\tag{9.3}
$$

Where, M_k and m_k form the alternating series of dominant input extrema, shown in Fig. 9.3a as the vertices that describe the staircase interface $L(t)$ between the sets S^+ and S^- (which correspond to $\hat{\gamma}$ operators in the "up" and "down" position respectively), briefly described in appendix 9.3 and in more detail in Chapter 1 of [17].



(a) $\alpha\beta$ plane for monotonically decreasing input. [17].

(b) $\alpha\beta$ plane for monotonically increasing input. [17].

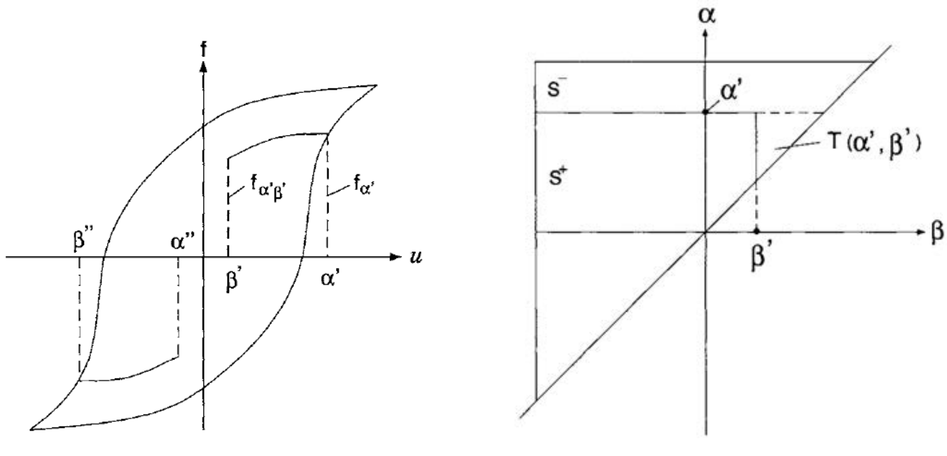
Figure 9.3: $\alpha\beta$ input evolution.

The f^+ term represents the positive saturation output value. The $f_{M_k m_k}$ terms are area calculations which are obtained as follows:

Towards identification, in order to relate the f function to the experimental data, the following function is defined [17]:

$$F(\alpha', \beta') = \frac{1}{2}(f_{\alpha'} - f_{\alpha'\beta'}) \tag{9.4}$$

Where α and β as shown in Fig. 9.4a, with $f_{\alpha'\beta'}$ a notation used for the output value on the transition curve attached to the limiting ascending branch at the point $f(\alpha')$ [17].



(a) Generalized input-output major loop.

(b) $\alpha - \beta$ at t when $u = \beta'$

Figure 9.4: Input-output and $\alpha - \beta$ equivalence.

Then, using the fact that, as seen from the above plots, the triangle $T(\alpha', \beta')$ is added to the negative set S^- and subtracted from the positive set S^+, it is found by Mayergoyz then that the Preisach model matches the output increments along the first-order transition curve if the function $\mu(\alpha, \beta)$ satisfies

$$f_{\alpha'\beta'} - f_{\alpha'} = -2 \int\int_{T(\alpha'\beta')} \mu(\alpha, \beta) d\alpha d\beta \tag{9.5}$$

Then by comparing equations 9.4 and 9.5 it follows:

$$F(\alpha'\beta') = \int\int_{T(\alpha'\beta')} \mu(\alpha,\beta)d\alpha d\beta \tag{9.6}$$

Now, looking in Fig. 9.3a, by further dividing the set S^+ into n number of trapezoids Q_k, then the integral can be expressed as

$$\int\int_{S^+(t)} \mu(\alpha,\beta)d\alpha d\beta = \sum_{k=1}^{n(t)} \int\int_{Q_k^+(t)} \mu(\alpha,\beta)d\alpha d\beta \tag{9.7}$$

Next, by expressing each trapezoid Q_k as a difference of two triangles $T(M_k m_{k-1})$, $T(M_k m_k)$ such that:

$$\int\int_{Q_k^+(t)} \mu(\alpha,\beta)d\alpha d\beta = \int\int_{T(M_k m_{k-1})} \mu(\alpha,\beta)d\alpha d\beta - \int\int_{T(M_k m_k)} \mu(\alpha,\beta)d\alpha d\beta \tag{9.8}$$

Then using equation 9.6 it follows:

$$\begin{aligned} F_{M_k m_{k-1}} &= \int\int_{T(M_k m_{k-1})} \mu(\alpha,\beta)d\alpha d\beta \\ F_{M_k m_k} &= \int\int_{T(M_k m_k)} \mu(\alpha,\beta)d\alpha d\beta \end{aligned} \tag{9.9}$$

With T the triangle defined from the S^+ and S^- sets in the α,β plane as shown in Fig. 9.3a

Substituting 9.9 to 9.8

$$\int\int_{Q_k^+(t)} \mu(\alpha,\beta)d\alpha d\beta = F_{M_k m_{k-1}} - F_{M_k m_k} \tag{9.10}$$

Next, with the numerical implementation in mind, equation 2.1 can be expressed as an addition and subtraction of the integral of $\mu(\alpha,\beta)$ over the $S^+(t)$ [17]

$$f(t) = -\int\int_T \mu(\alpha,\beta)d\alpha d\beta + 2\int\int_{S^+(t)} \mu(\alpha,\beta)d\alpha d\beta \tag{9.11}$$

With T, the limiting triangle as shown in fig. 9.3a.

From 9.6

$$\int\int_T \mu(\alpha,\beta)d\alpha d\beta = F(\alpha_0,\beta_0) \tag{9.12}$$

And then from 9.11, 9.12, 9.8, 9.10 and the fact that $m_n = u(t)$ (from Fig. 9.3a) it follows:

$$f(t) = -F(\alpha_0,\beta_0) + 2\sum_{k=1}^{n(t)-1} [F_{M_k m_{k-1}} - F_{M_k m_k}] + 2[F_{M_n m_{n-1}} - F_{M_n u(t)}] \tag{9.13}$$

Finally, using the formula of 9.4 upon the above equation and repeating the same procedure but this time for a monotonically increasing input, the numerical versions of the preisach output functions f from equation 9.3 are obtained.

9.6 Power-Law digital implementation

9.6.1 mPL process

Following then are the two code snippets responsible of computing the model state at each sample n for N number of cells:

```
% Calculate voltage across each cell's capacitor by solving the
% circuit equations using Euler Integration method
for k = 1:N
     for n = 1:length(e)-1
         % euler
         u(k,n+1) = u(k,n) + ...
             NLRe_s(e(n)-u(k,n),Vt(k),Is(k))*dt;   % i_k
     end
end

 % VCVSs weighting
 wUk = w(2:end).*u';

 % Voltage across the linear resistor, by setting R=1, Equation [5.2]
 y = w(1)*e + sum(wUk',1) - Vb;
```

```
function I = NLRe_s(vk,Vt,Is)
    % Equation [5.1]
    I = Is*exp(vk/Vt)-Is*exp(-vk/Vt);
end
```

9.6.2 mPL compensator process

Following then are the two code snippets responsible of computing the compensator state at each sample n for N number of cells:

```
l = length(y);
u = zeros(N,l);
e = zeros(1,l);
for n = 1:l
    % Moving from right to left, start by solving the right part.
    wUk = w(2:end).*u(:,n)';

    e(n) = (1/w(1).*y(n)) - (1/w(1).*sum(wUk',1)) + (Vb/w(1));

    % now use the computed x from the right to solve the circuits on the left.
    if n ~=l
        for k = 1:N
            % euler
            u(k,n+1) = u(k,n) + ...
                NLRe_s(e(n)-u(k,n),Vt(k),Is(k))*dt;   % i_k
        end
    end
end
```

```
% apply compensator on second dataset with sensor response as input
[pos_comp,~] = PLcompensator(time2,sens_out2,par,w_fit,weight_loc);
```

9.7 SMAC excitation

The SMAC actuator is controlled in position-control mode under a sine-wave excitation of 0.5 Hz and an amplitude of 3 mm.

$$\begin{aligned} x_{\text{smac}}(t) &= O + A_{\text{smac}} \cdot \sin(2\pi f t) \\ x_{\text{smac}}(t) &= 50 + 600\sin(\pi t) \end{aligned} \tag{9.14}$$

Where, the offset O (distance from zero point) and amplitude A are expressed in unitless encoder increments. With a 5 µm encoder resolution, a 3 mm amplitude is expressed as

$$A_{\text{smac}} = \frac{3}{5} = \frac{[\text{mm}]}{[\text{um}]} = 600 \tag{9.15}$$

Although, based on the most current (regarding this assignment) calibration, the relation between the stroke displacement and the encoder can be expressed with the following linear relation

$$y = 0.0051x - 0.0656 \tag{9.16}$$

Where y is the stroke displacement in mm and x is the encoder points. Based on this calibration the produced amplitude is

$$\hat{A}_{\text{smac}} = 0.0051 \cdot 600 - 0.0656 = 3.1256\ \text{mm} \tag{9.17}$$

9.8 Half-bridge readout

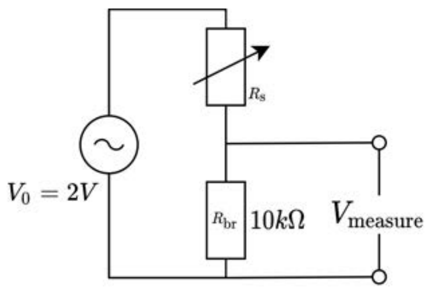


Figure 9.5: Half-Bridge (Voltage-Divider) configuration for a single strain-gauge measurement.

9.9 MATLAB-Ansys Link

Towards the technical issue of communication between MATLAB and Ansys, due to the wide use of both software packages, the desire to use a combined simulation has been an object of several studies. Two main solutions are found:

1 MATLAB AAS Toolbox.

2 Interface through batch files (Windows) directly to Ansys core (APDL).

3 Community provided packages [53, 54] in combination with [2].

Option 1 is the most straightforward and requires the least amount of implementation. The link between the two packages is established with the additional toolbox and examples of configuration for both ends are available within toolbox documentation. The only drawback in the specific case of the current work, is that the toolbox is not accessible through the provided Ansys client.

Option 2 requires more effort, basic knowledge of batch commands in Windows OS, the knowledge of the APDL scripting interface of Ansys and from scratch development of the link between the two packages.

Option 3 combines examples already available from the MATLAB community with option 2, minimizing that way the effort required for a manual setup of the link.

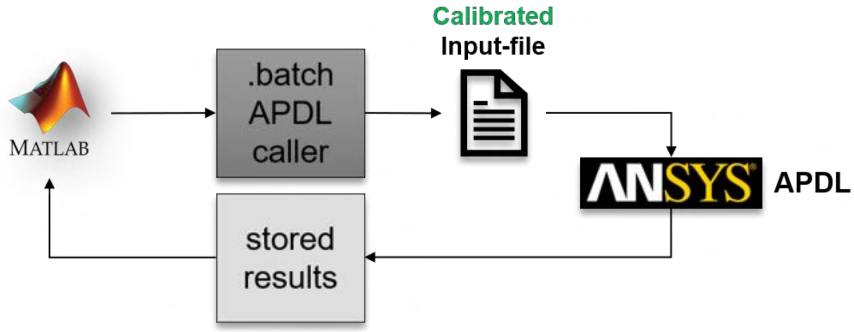


Figure 9.6: Dual-simulation scheme.

The toolbox would be the suggested solution to almost all scenarios, but the fact that it is not available for the current project renders the option impossible to follow. Therefore the third option is the remaining alternative.

Drawback

Both option 2 & 3, as currently planned, do not receive output from the running Ansys FEA solver. For the latter, the toolbox is necessary. Given the toolbox a true co-simulation is possible. With the other options, the APDL commands passed via a batch file can accept arguments where the excitation signal should be given as whole. Therefore, these alternative result in a dual-simulation, instead of a co-simulation.

Towards the implementation, given the entire set-up for the non-linear model calibration, the majority of the work is already present. Specifically:

- APDL command script of the entire simulation (Input-file).
- A .batch file responsible for calling the APDL core in batch mode and passing the current iteration arguments.

The last step is the MATLAB implementation for calling the batch file and reading from the stored results. Both of the latter are solved problems which minimal technical complexity. An overview of this scheme is given in Fig. 9.6.

Bibliography

[1] I. Mayergoyz and G. Friedman, "Generalized preisach model of hysteresis," *IEEE Transactions on Magnetics*, vol. 24, no. 1, pp. 212–217, 1988.

[2] L. Chua and K. Stromsmoe, "Lumped-circuit models for nonlinear inductors exhibiting hysteresis loops," *IEEE Transactions on Circuit Theory*, vol. 17, no. 4, pp. 564–574, 1970.

[3] "A pwl ladder circuit which exhibits hysteresis," vol. 22.

[4] A. Oliveri, F. Stellino, G. Caluori, M. Parodi, and M. Storace, "Open-Loop Compensation of Hysteresis and Creep Through a Power-Law Circuit Model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 3, pp. 413–422, 2016.

[5] A. Oliveri, F. Stellino, M. Parodi, and M. Storace, "Hysteresis and creep: Comparison between a power-law model and Kuhnen's model," *Physica B: Condensed Matter*, vol. 486, pp. 2–6, 2016.

[6] SMAC Moving Coil Actuators, "Embedded motion control library," Revision 1.3 (Firmware version 2.0). 2013 INGENIA-CAT, S.L.

[7] Florian Knorn, "Arrowhead," Version 1.0.0 (R2019b). MATLAB Central File Exchange. Retrieved September 30, 2020.

[8] Veryst Engineering, "Polyumod, a library of advanced user materials," Version 5.3.0. Veryst Engineering, LLC.

[9] M. Schouten, D. Kosmas, and G. Krijnen, "Hysteresis compensation of 3d printed sensors using a power law model for various input signals," *2020 IEEE SENSORS*, 2020.

[10] A. Elliott and C. Waters, *Additive manufacturing for designers.* SAE International.

[11] M. Rafiee, R. D. Farahani, and D. Therriault, "Multi-material 3d and 4d printing: A survey," *Advanced Science*, vol. 7, no. 12, p. 1902307, 2020.

[12] H. J. Pandya, J. Sheng, and J. P. Desai, "Mems-based flexible force sensor for tri-axial catheter contact force measurement," *Journal of Microelectromechanical Systems*, vol. 26, no. 1, pp. 264–272, 2017.

[13] S. Tung, S. R. Witherspoon, L. A. Roe, A. Silano, D. P. Maynard, and N. Ferraro, "A mems-based flexible sensor and actuator system for space inflatable structures," *Smart Materials and Structures*, vol. 10, no. 6, pp. 1230–1239, 2001.

[14] M. Schouten, B. Prakken, R. Sanders, and G. Krijnen, "Linearisation of a 3d printed flexible tactile sensor based on piezoresistive sensing," in *2019 IEEE SENSORS*, pp. 1–4, Oct 2019.

[15] M. Biggio, A. Oliveri, F. Stellino, M. Parodi, and M. Storace, "A circuit model of hysteresis and creep," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 5, pp. 501–505, 2015.

[16] F. Preisach, "Uber die magnetische nachwirkung," *Zeitschrift fur Physik*, vol. 94, no. 5-6, pp. 277–302, 1935.

[17] I. D. Mayergoyz, *Mathematical models of hysteresis and their applications.* Elsevier, 2003.

[18] H. Qi and M. Boyce, “Stress-strain behavior of thermoplastic polyurethanes,” *Mechanics of Materials*, vol. 37, no. 8, pp. 817–839, 2005.

[19] S. Bobbio and G. Marrucci, “A possible alternative to preisach’s model of static hysteresis,” *Il Nuovo Cimento D*, vol. 15, no. 5, pp. 723–733, 1993.

[20] J. A. Barker, D. E. Schreiber, B. G. Huth, and D. H. Everett, “Magnetic hysteresis and minor loops: Models and experiments,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 386, no. 1791, pp. 251–261, 1983.

[21] M. Parodi, M. Storace, and S. Cincotti, “Static and dynamic hysteretic features in a pwl circuit,” *International Journal of Circuit Theory and Applications*, vol. 24, no. 2, pp. 183–199, 1996.

[22] M. Storace and M. Parodi, “On the representation of static hysteresis curves by a pwl ladder circuit,” *International Journal of Circuit Theory and Applications*, vol. 26, no. 2, pp. 167–177, 1998.

[23] K. Kuhnen and H. Janocha, “Operator-Based Compensation of Hysteresis, Creep and Force-Dependence of Piezoelectric Stack Actuators,” *IFAC Proceedings Volumes*, vol. 33, no. 26, pp. 407–412, 2000.

[24] P. Krejci and K. Kuhnen, “Inverse control of systems with hysteresis and creep,” *University of Manitoba*, p. 144, 2001.

[25] K. Kuhnen, “Modeling, identification and compensation of complex hysteretic nonlinearities: A modified prandtl-ishlinskii approach,” *European Journal of Control*, vol. 9, no. 4, pp. 407–418, 2003.

[26] K. Kuhnen, “Modelling, identification, and compensation of complex hysteretic and log(t)-type creep nonlinearities,” *Control and Intelligent Systems*, vol. 33, no. 2, pp. 134–147, 2005.

[27] K. Kuhnen and P. Krejci, “Compensation of complex hysteresis and creep effects in piezoelectrically actuated systems - A new Preisach modeling approach,” *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 537–550, 2009.

[28] M. A. Krasnosel’skiĭ and A. V. Pokrovskiĭ, *Systems with Hysteresis.* Springer, 1989.

[29] V. Hassani, T. Tjahjowidodo, and T. N. Do, “A survey on hysteresis modeling, identification and control,” *Mechanical Systems and Signal Processing*, vol. 49, no. 1-2, pp. 209–233, 2014.

[30] A. Oliveri, M. Lodi, F. Stellino, and M. Storace, “Modeling and compensation of hysteresis and creep: The HysTool toolbox,” *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2018-May, no. May, 2018.

[31] Q. Xu, “Identification and compensation of piezoelectric hysteresis without modeling hysteresis inverse,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 3927–3937, 2013.

[32] A. J. Fleming, “Charge drive with active dc stabilization for linearization of piezoelectric hysteresis,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 60, no. 8, pp. 1630–1637, 2013.

[33] Y. Liu, J. Shan, U. Gabbert, and N. Qi, “Hysteresis and creep modeling and compensation for a piezoelectric actuator using a fractional-order maxwell resistive capacitor approach,” *Smart Materials and Structures*, vol. 22, no. 11, p. 115020, 2013.

[34] B. Mokaberi and A. Requicha, “Compensation of scanner creep and hysteresis for afm nanomanipulation,” *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 2, pp. 197–206, 2008.

[35] “Matlab optimization toolbox,” Version 8.4 (R2019b). MathWorks, Natick, MA, USA.

[36] M. Schouten, G. Wolterink, A. Dijkshoorn, D. Kosmas, S. Stramigioli, and G. Krijnen, “A Review of Extrusion Based 3D Printing for the Fabrication of Electro- and Biomechanical Sensors,” *Special issue on 20 Years IEEE sensors Journal, August 2020*, vol. 2020-August, 2020.

[37] Palmiga Innovation, “rubber3dprinting.com.” url: https://rubber3dprinting.com/pi-etpu-95-250-carbon-black/, Accessed: 10-05-2020.

[38] Fenner Drives, “Eel 3d printing filament,” report, 2018.

[39] Fenner Drives, “Ninjatek.com.” url: https://ninjatek.com, Accessed: 22-02-2020.

[40] eSUN, “esun3d.net.” http://www.esun3d.net/products/82.html, Accessed: 10-05-2020.

[41] Ultimaker, “ultimaker.com.” url: https://ultimaker.com/software/ultimaker-cura, Accessed: 10-05-2020.

[42] Schouten Martijn, “Nifty lab equipment drivers.”

[43] ONSEMI [ON Semiconductor], “Mc74act04n hex inverter.”

[44] Pico Technology, “Picoscope 5000d series user’s guide.”

[45] TiePie Engineering, “Handyscope hs5 user manual.”

[46] Eric Perez Lorea, “3d printed sensor to detect muscle contraction by means of force myography,” 2017. University of Twente, Robotics and Mechatronics.

[47] A. Oliveri, M. Lodi, F. Stellino, and M. Storace, “Modeling and compensation of hysteresis and creep: The hystool toolbox,” *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018.

[48] J. Bergstrom, “Determination of material parameters for the 8-chain model for use with abaqus, ls-dyna and ansys,” 2020.

[49] Jorgen Bergstrom, “Bergostrom-boyce model for thermoplastics.”

[50] “Control system toolbox - design and analyze control systems,” Version 8.4 (R2019b). MathWorks, Natick, MA, USA.

[51] “Robotics system toolbox,” Version 8.4 (R2019b). MathWorks, Natick, MA, USA.

[52] E. Schrijver and J. van Dijk, “Disturbance observers for rigid mechanical systems: Equivalence, stability, and design,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, no. 4, pp. 539–548, 2002.

[53] Mikkel Pedersen, “Ansysimport,” Version 1.0.0 (R2019b). MATLAB Central File Exchange. Retrieved October 7, 2020.

[54] Nick, “Interactive link between matlab and ansys,” Version 1.0.0 (R2019b). MATLAB Central File Exchange. Retrieved October 7, 2020.