RIJKSUNIVERSITEIT GRONINGEN

ON THE IDENTIFICATION OF HYSTERETIC DEFORMABLE MIRRORS USING KERNEL METHODS

MASTER RESEARCH PROJECT IN APPLIED MATHEMATICS

Author:
Lisa Oosterhof

Student number:
S3032558

First Supervisor:
Henk van Waarde

Second Assessor:
Bayu Jayawardhana

September 19, 2023

university of groningen

faculty of science and engineering

Abstract

The hysteretic deformable mirror, abbreviated as HDM, is a deformable mirror with hysteresis behaviour. The HDM can be used in a telescope which can be used to study the properties of exoplanets. To identify the behaviour of the HDM, phenomenological models will be studied. To obtain such models from data, a suitable identification method has to be chosen. As the classical system identification approaches have many disadvantages such as overfitting, another approach will be taken. This research introduces regularized kernel methods in which the number of basis functions and data pairs is not fixed a priori. These methods are applied to obtain models of deformable mirrors that avoid overfitting, while also complying with the physics of the system.

Contents

1 Introduction

"Are we alone in this universe?" This is a question that has been one of the main topics in astronomy over the past years. The first exoplanets were already discovered in the 1990s. Nowadays, more than 4000 exoplanets have been detected. The techniques used to identify the exoplanets do not collect light from the planet itself but observe the effect it has on its accompanying star. The exoplanet is thus detected indirectly. It is a great challenge to find the exoplanet directly since the light of the planet is so much less bright than the light of its accompanying star.

After discovering a new exoplanet, one would like to know all information about the chemical composition of a possible atmosphere surrounding the planet. This information is necessary to say something about the possibility of life on the planet. However, with the used indirect method, it is not possible to observe the chemical composition of the possible atmosphere of the discovered planet. Therefore a direct method to observe exoplanets is needed.

To directly detect the planet, the contrast in brightness of the light of the exoplanet and the accompanying star has to be resolved. This is needed to avoid the star from completely drowning out the light from the planet. This requires the application of a coronagraph, starshade, or even a space interferometer. Future large space telescopes that will directly image exoplanets will rely on deformable mirrors [7].

The mirrors should be able to deform in order to detect the planet directly. This requires specific types of materials. One example is a piezoelectric material. Piezoelectric material has a lot of desired properties. Piezoelectric-based actuators can exhibit hysteresis. Such a mirror is called a hysteretic deformable mirror, abbreviated as HDM. The goal of the HDM is to achieve and hold a desired surface deflection for a long period of time where an individual constant control for every actuator is not needed [2].

Exploring the topic of deformable mirrors is interdisciplinary research. The disciplines of astronomy, advanced materials, engineering, physics, chemistry, and mathematics work together. In this research, the focus is on applying mathematics to the deformable mirror. Before being able to describe the behaviour of the deformable mirror, a mathematical model has to be identified.

A classical approach to identify a mathematical model is the prediction error method. By minimizing the errors of the predicted values [3], this method tries to find an optimal solution. Another approach is the machine learning approach. Within the machine learning approach, kernel methods will be used. The optimal solution within kernel methods can be achieved by transforming the given input into a higher-dimensional space by making use of kernel functions. This allows linear algorithms to identify a system [4].

The organization of this research is as follows. In Chapter 2, an overview of hysteresis behaviour is given. General notions, different operators, and some examples will be explored. In Chapter 3, the identification is studied. This consists of both the classical method and the kernel method. Chapter 5 includes an experiment that has been performed. After performing the experiment, the hysteresis behaviour of the HDM was studied and the identification of this HDM is included.

2 Hysteresis behaviour

In 1885 the term hysteresis was formulated by Ewing to describe a physical phenomenon that occurred in the magnetization process of soft iron. This magnetization process was caused by recurring changes in the input magnetic field. The term hysteresis is derived from the Ancient Greek word υστερησις meaning "delay" [19]. Nowadays, hysteresis is a phenomenon that is present in systems such as the piezo-actuator, ferromagnetic material and mechanical systems [13]. Hysteresis is a nonlinear function with memory which cannot be described by a single-valued function. To be able to describe hysteretic phenomena, different hysteresis models have been developed [14]. Two distinctive hysteresis models are defined. First of all, the physics-based models describe the hysteresis phenomenon from particular physical relations and second, the phenomenological model focuses on the empirical description of input-output behaviour. Due to simplicity and completeness, phenomenological models have been widely studied. The phenomenological models that are used in general are the Preisach model and the Duhem hysteresis operator.

2.1 Operators

In pursuance of understanding the models, first, some properties of operators have to be defined. Let $\Phi : \mathscr{C}_a(\mathbb{R}_+, \mathbb{R}) \times \mathbb{R} \to \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ be an operator, where $\mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ is the space of absolutely continuous functions. Let one consider the following definition for rate-independency [3].

Definition 1. An operator is rate-independent if for every $\phi \in \mathscr{C}(\mathbb{R}_+, \mathbb{R}_+)$, where $\mathscr{C}(\mathbb{R}_+, \mathbb{R}_+)$ is the space of continuous functions, such that $\phi(0) = 0$, increasing and radially unbounded, meaning that $\lim_{t\to\infty} \phi(t) = \infty$, we have

$$[\Phi(u \circ \phi)](t) = [\Phi(u)] \circ \phi(t)$$

holds for all $u \in \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ and for all admissible time transformation ϕ.

Next, causality will be defined [3].

Definition 2. An operator Φ is causal if $\forall\ \tau > 0$ and $\forall\ u_1, u_2 \in \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$, it holds that

$$u_1(t) = u_2(t)\ \ \forall\ t \in [0, \tau] \implies [\Phi(u_1)](t) = [\Phi(u_2)](t)\ \ \forall\ t \in [0, \tau].$$

When both casualty and rate-independency are considered, a specific type of operator is obtained [3]

Definition 3. An operator Φ is called a hysteresis operator if it is both causal and rate-independent.

The hysteresis loop is defined as the periodic closed orbit in the input-output phase plot that occurs when a periodic input with a single minimum and a single maximum is applied to a hysteresis operator. The mathematical definition is the following. Let us consider a hysteresis operator Φ and an input–output pair $(u, y) \in \mathscr{C}_a(\mathbb{R}_+, \mathbb{R}) \times \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ with $y = \Phi(u)$. Suppose u is periodic with period $\bar{T} > 0$. Define the maximum by $u_{\max} \in \mathbb{R}$ and the minimum in the periodic interval by $u_{\min} \in \mathbb{R}$. Assume that y is periodic in the interval $[c, \infty)$.

Definition 4. The periodic orbit denoted by $\mathcal{H}_{u,y} = \{(u(t), y(t)|t \in [c, \infty)\}$ is called a hysteresis loop if there exists $\nu \in \mathbb{R}$ such that the cardinality of the set $\{(\nu, \gamma) \in \mathcal{H}_{u,y} \mid \gamma \in \mathbb{R}\}$ is equal to 2.

This implies that the curve $\mathcal{H}_{u,y}$ can have for any admissible point v at most two elements (v, γ_1) and (v, γ_2).

To define the input-output behaviour of hysteresis operators, the type of hysteresis loops reproduced by the operators have to be studied. When one considers a simple hysteresis loop, meaning that it consists of only one hysteresis loop, it can either have a clockwise or counterclockwise orientation. A hysteresis operator exhibits clockwise input-output behaviour if there exists at least one hysteresis loop $\mathcal{H}_{u,y}$ which corresponds to an input-output pair (u, y) with $y = \Phi(u)$ which is clockwise. The signed area enclosed by an input-output pair (u, y) that forms a closed curve in a given interval from t_1 to t_2 is, by Green's Theorem, defined as

$$\mathcal{A} = \frac{1}{2}\int_{t_1}^{t_2} [u(\tau)\dot{y}(\tau) - y(\tau)\dot{u}(\tau)]d\tau.$$

If $t_1 > t_p$, $t_2 = t_1 + T$ and the signed-area by an input-output pair (u, y) that forms a closed curve in an interval $[t_1, t_2]$ given by $\mathcal{A}$ satisfies $\mathcal{A} < 0$, then the loop is clockwise. Similar reasoning can be followed to specify the counterclockwise orientation.

As its name already suggests, butterfly loops consist of two subloops connected by a self-intersection point that has the shape of a butterfly. One loop is oriented clockwise and the other is counterclockwise oriented. The total signed area depends on the difference between the signed area of the clockwise-oriented subloop and the counterclockwise-oriented subloop.

Definition 5. An operator Φ is called a butterfly hysteresis operator if there exists a hysteresis loop for which the signed area of the clockwise oriented loop equals the signed area of the counterclockwise oriented loop.

The butterfly hysteresis operator only requires a single input-output pair for which $\mathcal{A} = 0$, there are no restrictions on the other input-output pairs. An example of a hysteresis operator is the relay operator.

Example 1. There is both a clockwise and counterclockwise relay operator. First of all the counterclockwise relay operator given by $\mathcal{R}^{\circlearrowleft}_{\alpha,\beta} : \mathscr{C}_a(\mathbb{R}_+, \mathbb{R}) \times \{-1, 1\} \to \mathscr{C}_p(\mathbb{R}_+, \mathbb{R})$ where α and β are the so-called switching values, those parameters can be varied. The space $\mathscr{C}_p(\mathbb{R}_+, \mathbb{R})$ is the space of piecewise continuous functions. The initial condition at $t = 0$ is given by r_0. The counterclockwise relay operator is defined as

$$[\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_0)](t) = \begin{cases} 1, & \text{if } u(t) > \alpha \\ -1, & \text{if } u(t) < \beta \\ [\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_0)](t_-), & \text{if } \alpha \geqslant u(t) \geqslant \beta \text{ and } t > 0 \end{cases}.$$

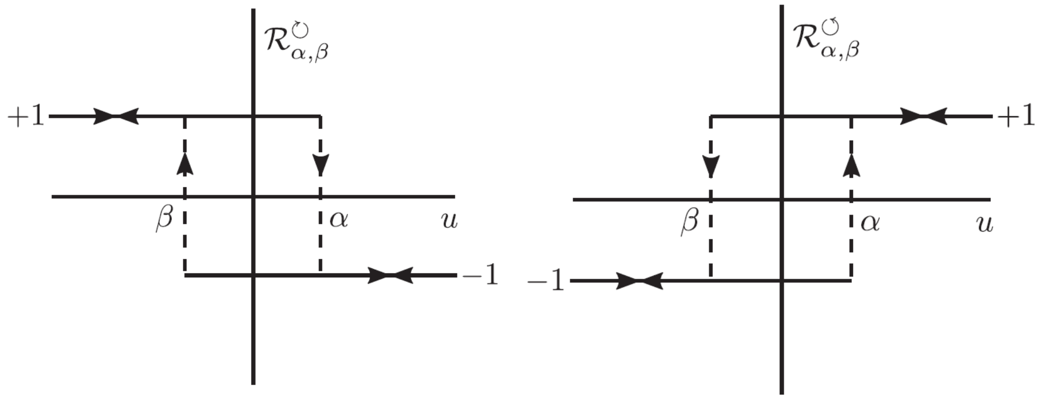
Similarly, the clockwise relay operator is defined as $\mathcal{R}^{\circlearrowright}_{\alpha,\beta} : \mathscr{C}_a(\mathbb{R}_+, \mathbb{R}) \times \{-1, 1\} \to \mathscr{C}_p(\mathbb{R}_+, \mathbb{R})$ by

$$[\mathcal{R}^{\circlearrowright}_{\alpha,\beta}(u, r_0)](t) = \begin{cases} -1, & \text{if } u(t) > \alpha \\ 1, & \text{if } u(t) < \beta \\ [\mathcal{R}^{\circlearrowright}_{\alpha,\beta}(u, r_0)](t_-), & \text{if } \alpha \geqslant u(t) \geqslant \beta \text{ and } t > 0 \end{cases}.$$

For a specified initial condition $r_0 \in \{-1, 1\}$ both relay operators are hysteresis operators of the form

$$\Phi(u) = \mathcal{R}^{\circlearrowright}_{\alpha,\beta}(u, r_0) \qquad \Phi(u) = \mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_0).$$

Note that $\mathcal{R}^{\circlearrowright}_{\alpha,\beta} = -\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}$. The input-output phase plots of the hysteresis operators are shown in Figure 1 [19].


$\mathcal{R}^{\circlearrowright}_{\alpha,\beta}$
+1
β
α
u
−1
$\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}$
+1
β
α
u
−1

(a) Clockwise relay operator. (b) Counterclockwise relay operator.

Figure 1: Input-output phase plot of the clockwise and counterclockwise operator.

Dependent on what type of hysteresis loops are produced by the hysteresis operators, the input-output behaviour of hysteresis operators can be classified [2]. Two phenomenological models that are used to describe the hysteresis behaviour are the Duhem hysteresis operator and the Preisach operator.

2.2 Preisach Operator

The Preisach operator is the weighted integral of all infinitesimal weighted relay operators for which the switching values satisfy $\alpha > \beta$. This operator $\mathcal{P} : \mathscr{C}_a(\mathbb{R}_+, \mathbb{R}) \times \mathcal{I} \to \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ can be expressed as

$$[\mathcal{P}(u, L_0)](t) := \iint\limits_{(\alpha,\beta)\in P} \mu(\alpha, \beta)[\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_{\alpha,\beta}(L_0))](t)d\alpha d\beta.$$

In this expression, P is the Preisach plane and is the admissible plane of switching values defined by $P := \{(\alpha, \beta) \in \mathbb{R}^2 | \alpha > \beta\}$. The set of interfaces given by $\mathcal{I}$ consists of interfaces L which is a monotonically decreasing staircase curve, which is parameterized by a function $\sigma(\gamma) \in \mathscr{C}(\mathbb{R}_+, P)$ which satisfies $\lim_{\gamma\to\infty} ||\sigma(\gamma)|| = \infty$ and $\sigma(0) = (\alpha, \alpha)$ for some $\alpha \in \mathbb{R}$ in the form $L = \{\sigma(\gamma) \in P | \gamma \in \mathbb{R}_+\}$ [3]. The staircase curve is given by

$$S(x, A, T) = A\left(-\frac{1}{2} + \frac{x}{T} + \frac{1}{\pi}\left(\cot\frac{\pi x}{T}\right)\right),$$

where A is the distance of the consecutive steps and T is the horizontal distance between two consecutive risers [5]. Furthermore, $\mu \in \mathscr{C}_p(P, \mathbb{R})$ is a weighting function, $L_0 \in \mathcal{I}$ is the initial interface and finally $r_{\alpha,\beta}(L_0)$ determines the initial condition of every relay $\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}$. When the point $(\alpha, \beta) \in P$ is above the initial interface L_0, $\mathcal{R}^{\circlearrowleft}_{\alpha,\beta} = 1$. When this point is below the initial interface, $\mathcal{R}^{\circlearrowleft}_{\alpha,\beta} = -1$. For well-posedness in the sense that a solution exists and only varies a bit if the initial conditions are changed, it is assumed that the initial interface given by L_0 in the Preisach operator satisfies $[u(0), u(0)] \in L_0$. Just as the example in Figure 1, also the Preisach operator is a hysteresis operator and defined by $\Phi(u) = \mathcal{P}(u, L_0)$ with specified initial conditions $L_0 \in \mathcal{I}$. In general, it can be said that the output of the Preisach operator is determined instantly with the variations of the given input since all the relays react simultaneously and at the same time to the applied input.

2.3 Preisach Butterfly Operator

Now both the Preisach operator and the hysteresis butterfly operator is studied, the Preisach butterfly operator can be defined. In the Preisach operator, only a positive weighting function together with counterclockwise loops is considered. When the weighting function is negative and the relays are still counterclockwise, the hysteresis loops become clockwise which is because $\mathcal{R}^{\circlearrowleft}_{\alpha,\beta} = -\mathcal{R}^{\circlearrowright}_{\alpha,\beta}$. By considering a weighting function that can be both positive and negative, an operator exhibits both a clockwise and counterclockwise input-output behaviour. This is the general idea of a butterfly hysteresis operator. To define a Preisach butterfly operator, the following lemma first needs to be considered.

Lemma 1. *Consider the counterclockwise relay operator $\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}$ where $\alpha > \beta$. For every signal u which is periodic with time T and $u_{\min}, u_{\max} \in \mathbb{R}$ the minimum and maximal signal in the periodic interval, the signed-area $\mathcal{A}$ belonging to the input-output pair (u, y) with $y = \mathcal{R}^{\circlearrowright}_{\alpha,\beta}(u, r_0)$ with $r_0 \in \{-1, 1\}$ is given by*

$$\mathcal{A} = \begin{cases} 2(\alpha - \beta), & \textit{if } u_{\min} < \beta \textit{ and } \alpha < u_{\max} \\ 0, & \textit{otherwise} \end{cases}.$$

Note that when $u_{\min} < \beta$ and $\alpha < u_{\max}$, the signed-area enclosed by the hysteresis loop of a clockwise relay operator $\mathcal{R}^{\circlearrowright}_{\alpha,\beta}$ is given by $-2(\alpha - \beta)$.

With this lemma, the formal theorem of a Preisach butterfly operator can be stated.

Theorem 1. *Consider the Preisach operator denoted by $\mathcal{P}$ where μ is a weighting function that has to take both positive and negative values. Suppose that the first order lower and upper partial moments of μ satisfy*

$$\int_{r}^{\infty} \mu(\alpha, \beta)\beta d\beta = \infty$$

$$\int_{-\infty}^{r} \mu(\alpha, \beta)\alpha d\alpha = \infty,$$

$\forall(\alpha, \beta) \in \mathcal{P}$ *and the boundary curve is monotonically decreasing, then $\mathcal{P}$ is a butterfly hysteresis operator.*

The proof of this theorem can be found in [13]. To illustrate the butterfly-shaped behaviour, an example will be considered.

Example 2. Consider the input-behaviour when a periodic input u with $u_{\max} = 1$ and $u_{\min} = -1$ is applied to the butterfly hysteresis operator. There are five-time instances considered.

$$\begin{aligned} t_1 &= 1 \\ t_2 &= 2 \\ &\vdots \\ t_5 &= 5. \end{aligned}$$

The input satisfies the following conditions

$$\begin{aligned} u(t_1) &= -1 & \qquad u(t_4) &= 0 \\ u(t_2) &= 0 & \qquad u(t_5) &= -1 \\ u(t_3) &= 1 & & \end{aligned}$$

and is defined in the following way

$$\begin{aligned} u(t) &= t^2 - 2t, & t_1 &\leqslant t \leqslant t_2, \\ u(t) &= t^2 - 4t + 4, & t_2 &\leqslant t \leqslant t_3, \\ u(t) &= -t^2 + 6t - 8, & t_3 &\leqslant t \leqslant t_4, \\ u(t) &= -t + 4, & t_4 &\leqslant t \leqslant t_5. \end{aligned}$$

The graph of the input signal is given in Figure 2.

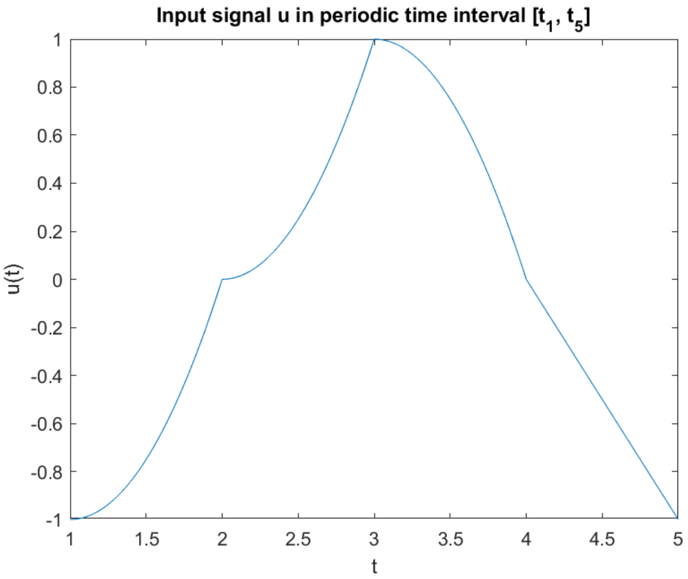


Figure 2: The defined input signal.

To define the output signal, the Preisach domain can be divided into four disjoint regions by

$$
\begin{aligned}
\mathcal{P}^1 &:= \{(\alpha,\beta)|\alpha \leqslant 0, \beta \leqslant 0\} \\
\mathcal{P}^2 &:= \{(\alpha,\beta)|\alpha > 0, \beta \leqslant 0, \alpha \leqslant -\beta\} \\
\mathcal{P}^3 &:= \{(\alpha,\beta)|\beta \leqslant 0, \alpha > -\beta\} \\
\mathcal{P}^4 &:= \{(\alpha,\beta)|\alpha > 0, \beta > 0\}
\end{aligned}
$$

The output can be computed by the behaviour of each region by adding and subtracting the integrals

$$
\begin{aligned}
\mathcal{P}(u, L_0)(t) = &- \iint\limits_{(\alpha,\beta)\in\mathcal{P}^1} \left[\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_{\alpha,\beta}(L_0))\right](t) d\alpha d\beta \right] - \iint\limits_{(\alpha,\beta)\in\mathcal{P}^2} \left[\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_{\alpha,\beta}(L_0))\right](t) d\alpha d\beta \right] \\
&+ \iint\limits_{(\alpha,\beta)\in\mathcal{P}^3} \left[\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_{\alpha,\beta}(L_0))\right](t) d\alpha d\beta \right] + \iint\limits_{(\alpha,\beta)\in\mathcal{P}^4} \left[\mathcal{R}^{\circlearrowleft}_{\alpha,\beta}(u, r_{\alpha,\beta}(L_0))\right](t) d\alpha d\beta \right]
\end{aligned}
$$

For the defined input, the output signal that belongs to the hysteresis operator is given by

$$[\mathcal{P}(u, L_0)](t) = \begin{cases} -[u_{\max} + u(t)]^2, & t_1 \leqslant t \leqslant t_2, \\ -[u_{\max} - u(t)][u_{\max} + 3u(t)], & t_2 \leqslant t \leqslant t_3, \\ -[u_{\max} - u(t)]^2, & t_3 \leqslant t \leqslant t_4, \\ -[u_{\max} + u(t)][u_{\max} - 3u(t)], & t_4 \leqslant t \leqslant t_5. \end{cases}$$

$$= \begin{cases} -[1 + u(t)]^2, & t_1 \leqslant t \leqslant t_2, \\ -[1 - u(t)][1 + 3u(t)], & t_2 \leqslant t \leqslant t_3, \\ -[1 - u(t)]^2, & t_3 \leqslant t \leqslant t_4, \\ -[1 + u(t)][1 - 3u(t)], & t_4 \leqslant t \leqslant t_5. \end{cases}$$

Then the input-output phase plot of the defined input and output signal can be created and can be found in Figure 3 [19].

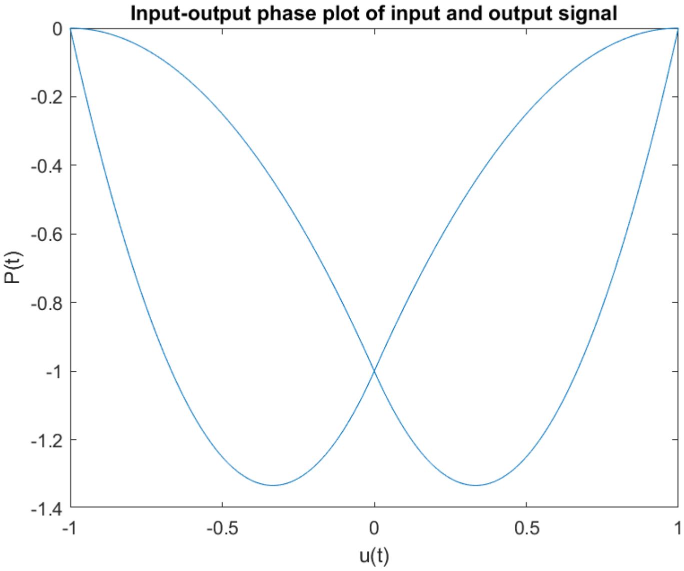


Figure 3: Input-output phase plot.

2.4 Duhem Operator

Next to the Preisach operator, another operator phenomenological model that is used to describe the hysteresis behaviour is the Duhem model. This model focuses on the fact that the output can only change its character when the direction of the input changes. The input signals are mapped to output signals via switched nonlinear differential equations [11]. The Duhem operator hysteresis operator is a mapping $\Psi : \mathscr{C}_a(\mathbb{R}_+, \mathbb{R}) \times \mathbb{R} \to \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ such that $y = \Psi(u, y_0)$ satisfies

$$\dot{y}(t) = \begin{cases} f_1(u(t), y(t))\dot{u}(t), & \text{if } \dot{u}(t) \geqslant 0, \\ f_2(u(t), y(t))\dot{u}(t), & \text{if } \dot{u}(t) < 0, \end{cases} \tag{1}$$

with $y(0) = y_0$, and functions $f_1, f_2 \in C^1(\mathbb{R}^2, \mathbb{R})$. There exists solutions to (1) if the following two conditions hold for every $\nu \in \mathbb{R}$

$$(\gamma_1 - \gamma_2)[f_1(\nu, \gamma_1) - f_1(\nu, \gamma_2)] \leqslant \lambda_1(\nu)(\gamma_1 - \gamma_2)^2,$$
$$(\gamma_1 - \gamma_2)[f_2(\nu, \gamma_1) - f_2(\nu, \gamma_2)] \geqslant -\lambda_2(\nu)(\gamma_1 - \gamma_2)^2,$$

with $\lambda_1, \lambda_2 \geqslant 0$ and $\gamma_1, \gamma_2 \in \mathbb{R}$ [14]. For the Duhem hysteresis operator, it is assumed that an implicit function $\nu \mapsto \{\gamma \in \mathbb{R} | f_1(\nu, \gamma) - f_2(\nu, \gamma) = 0\}$ takes an explicit solution given by $\gamma = \rho(\nu)$, with the anhysteresis function $\rho \in \mathscr{C}^0(\mathbb{R}, \mathbb{R})$. The curve generated by ρ is called the anhysteresis curve and is denoted by

$$\mathcal{P} = \{(\nu, \gamma) \in \mathbb{R}^2 | \gamma = \rho(\nu)\}.$$

The anhysteresis curve divides the input-output plane into two regions since the system's response to increasing input is different from its response to decreasing input. The two regions are given by:

$$\begin{cases} f_1(\nu, \gamma_1) - f_2(\nu, \gamma_1) \geqslant 0, & \text{if } \gamma_1 \geqslant \gamma, \\ f_1(\nu, \gamma_1) - f_2(\nu, \gamma_1) \leqslant 0, & \text{if } \gamma_1 \leqslant \gamma, \end{cases}.$$

The most important property of the Duhem operator Ψ is the accommodation property. Before starting the accommodation property, first, the Duhem operator for the periodic input must be defined. Let $u_p \in \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ be a periodic input with a certain period given by $T > 0$. This input has one minimum $u_p(0) = \nu_{\min}$ and one maximum $u_p(t_1) = \nu_{\max}$ for some $t_1 \in (0, T)$.

The periodic input u_p can be split up into two monotonic intervals which are given by u_{p-} and u_{p+}. The corresponding outputs are given by $Y_{u_{p-}}(t,\zeta)$ and $Y_{u_{p+}}(t,\gamma)$, where γ, ζ are the independent variables. Let $y_{u_{p+}}$ and $y_{u_{p-}}$ be the parameterization of the corresponding solutions $Y_{u_{p-}}(t,\zeta)$ and $Y_{u_{p+}}(t,\gamma)$ respectively. For $\gamma_0 \in \mathbb{R}$, let the two sequences $(\zeta_n)_{n\in\mathbb{N}}$ and $(\gamma_n)_{n\in\mathbb{N}}$ be defined recursively by

$$\begin{aligned} \zeta_n &:= y_{u_{p+}}(\nu_{\max},\gamma_n), \\ \gamma_{n+1} &:= y_{u_p}(\nu_{\min},\zeta_n). \end{aligned}$$

Let $\gamma_0 = y_0$, the output can also be recursively constructed:

$$Y_{u_p}(t,y_0) = \begin{cases} y_{u_{p+}}(u_p(t),\gamma_n), & \text{if } nT \leqslant t < t_1 + nT, \\ y_{u_{p-}}(u_p(t),\zeta_n), & \text{if } t_1 + nT \leqslant t < (n+1)T. \end{cases}$$

The accommodation property refers to the property where the input-output phase plot always converges to a periodic closed orbit given that the input signal is periodic. This property can be captured by two propositions. The first proposition, Proposition 1 shows a pair of inequalities that ensure the convergence to some periodic orbit. Proposition 2 shows that the strict versions of the inequalities in the first proposition ensure the uniqueness of the pair $\gamma^* \in \mathbb{R}$ and $\zeta^* \in \mathbb{R}$. Where γ^* and ζ^* are the pair to which the sequences converge to. By the continuity and uniqueness of the solution of the Duhem operator, the following equalities must hold:

$$\begin{aligned} y_{u_{p+}}(\nu_{\max},\gamma^*) &= \zeta^*, \\ y_{u_{p-}}(\nu_{\min},\zeta_n^*) &= \gamma^*. \end{aligned}$$

By this result, the parameterized solutions $y_{u_{p+}}(\nu,\gamma^*)$ and $y_{u_{p-}}(\nu,\zeta_n^*)$ form a closed orbit in the phase plot. Let us now state the first proposition.

Proposition 1. *If the functions f_1 and f_2 with f_1 and f_2 as in (1) satisfy $\forall \gamma_1 \neq \gamma_2$ and $\nu \in \mathbb{R}$*

$$\begin{aligned} (f_1(\nu,\gamma_1) - f_1(\nu,\gamma_2))(\gamma_1 - \gamma_2) &\leqslant 0, \\ (f_2(\nu,\gamma_1) - f_2(\nu,\gamma_2))(\gamma_1 - \gamma_2) &\geqslant 0, \end{aligned}$$

then $\forall \gamma_0 \in \mathbb{R}$ the sequences $(\zeta_n)_{n\in\mathbb{N}}$ and $(\gamma_n)_{n\in\mathbb{N}}$ are convergent.

Next to the convergence of solutions, one also wants to know whether those solutions are unique. If the inequalities in Proposition 1 are strict, then the solution is unique. This can be stated in the following proposition.

Proposition 2. *If the functions f_1 and f_2 with f_1 and f_2 as in (1) satisfy $\forall \gamma_1 \neq \gamma_2$ and $\nu \in \mathbb{R}$*

$$(f_1(\nu, \gamma_1) - f_1(\nu, \gamma_2))(\gamma_1 - \gamma_2) < 0,$$
$$(f_2(\nu, \gamma_1) - f_2(\nu, \gamma_2))(\gamma_1 - \gamma_2) > 0,$$

then $\exists$ a unique pair $\gamma^*, \zeta^* \in \mathbb{R}$ such that for every $\gamma_0 \in \mathbb{R}$, $(\gamma_n)_{n \in \mathbb{N}} \rightarrow \gamma^*$ and $(\zeta_n)_{n \in \mathbb{N}} \rightarrow \zeta^*$ with $\zeta^* = y_{u_{p+}}(\nu, \gamma^*)$

The proof of these propositions can be found in [3] and relies on the three following lemmas. For Proposition 1, the convergence has to be studied. Since the output recursively can be constructed, the convergence of the sequences has to be considered. First of all, both sequences are either increasing or decreasing. This is captured in Lemma 2

Lemma 2. *Let $\gamma_0 \in \mathbb{R}$. The sequences $(\zeta_n)_{n \in \mathbb{N}}$ and $(\gamma_n)_{n \in \mathbb{N}}$ are monotonic and either both increasing or both decreasing.*

The second important lemma tells us something about two consecutive elements in the sequence when they are the same.

Lemma 3. *Let $\gamma_0 \in \mathbb{R}$ and consider the sequences $(\zeta_n)_{n \in \mathbb{N}}$ and $(\gamma_n)_{n \in \mathbb{N}}$, then*

(a) If $\gamma_i = \gamma_{i+1}$ for some $i \in \mathbb{N}$, then $\forall k \geqslant i$,

$$\zeta_k = \zeta_{k+1} \quad \text{and} \quad \gamma_{k+1} = \gamma_{k+2}$$

(b) If $\zeta_j = \zeta_{j+1}$ for some $j \in \mathbb{N}$, then $\forall k \geqslant j$,

$$\zeta_{k+1} = \zeta_{k+2} \quad \text{and} \quad \gamma_{k+1} = \gamma_{k+2}$$

The final lemma shows that when both sequences do not have a bound, they are strictly monotonic in the same direction.

Lemma 4. *Let $\gamma_0 \in \mathbb{R}$ and consider the sequences $(\zeta_n)_{n \in \mathbb{N}}$ and $(\gamma_n)_{n \in \mathbb{N}}$ The sequence $(\zeta_n)_{n \in \mathbb{N}}$ is unbounded if and only if $(\gamma_n)_{n \in \mathbb{N}}$ is unbounded. Moreover, if they are both unbounded, they are strictly monotonic in the same direction.*

With the use of lemmas 2, 3 and 4 the accommodation property can be shown. As the existence and uniqueness of solutions have been stated, an example of the Duhem operator can help to illustrate the behaviour of the operator. The example chosen is the Bouc-Wen model. This model is commonly used to describe relations between displacement as input and restoring force as output in piezo-actuated mechanical systems. The Bouc-Wen model can be defined by the following differential equation

$$\dot{y}(t) = \alpha\dot{u}(t) - \beta|y(t)|^n\dot{u}(t) - \zeta y(t)|y(t)|^{n-1}|\dot{u}(t)|, \tag{2}$$

where $\alpha, \beta, \zeta \in \mathbb{R}$ are the model parameters.

Example 3. If $\alpha = \zeta = 1$ and $\beta = 2$, the condition for convergence as in Proposition 1 is satisfied. This differential equation can be rewritten in the form as in Equation 1 in the following manner

$$\begin{aligned}\dot{y}(t) &= \begin{cases} f_1(\nu,\gamma)\dot{u}(t), & \text{if } \dot{u}(t) \geqslant 0,\\ f_2(\nu,\gamma)\dot{u}(t) & \text{if } \dot{u}(t) < 0. \end{cases}\\ &= \begin{cases} (\alpha - \beta|\gamma|^n - \zeta\gamma|\gamma|^{n-1})\dot{u}(t), & \text{if } \dot{u}(t) \geqslant 0,\\ (\alpha - \beta|\gamma|^n + \zeta\gamma|\gamma|^{n-1})\dot{u}(t), & \text{if } \dot{u}(t) < 0. \end{cases}\\ &= \begin{cases} (1 - 2|\gamma|^n - \gamma|\gamma|^{n-1})\dot{u}(t), & \text{if } \dot{u}(t) \geqslant 0,\\ (1 - 2|\gamma|^n + \gamma|\gamma|^{n-1})\dot{u}(t), & \text{if } \dot{u}(t) < 0. \end{cases}\end{aligned}$$

To show this example, a periodic input function has to be considered. The input considered is the function $u(t) = \sin(t)$. The derivative of the input is given by The derivative of the input is defined in the following way $\dot{u}(t) = \cos(t)$. Using MATLAB's built-in function ODE45, the ordinary differential equation can be solved. The hysteresis loop obtained from the Bouc-Wen hysteresis operator when a periodic input is applied can be found in Figure 4

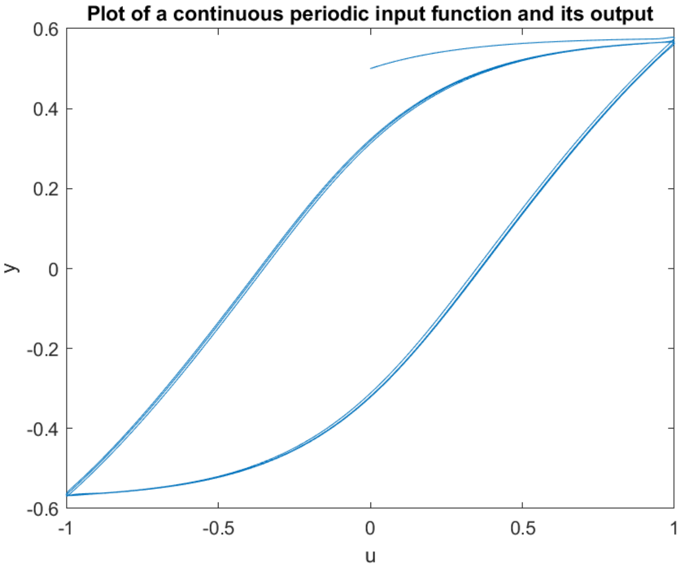


Figure 4: The Bouc-Wen Hysteresis Operator.

2.5 Duhem Butterfly Operator

A special class of Duhem operators is the Duhem butterfly operators. The principle of this operator is the same as for the Preisach butterfly operator. The operator can produce complex periodic hysteresis loops where it intersects itself. In this class of operators, f_1 and f_2 can both assume positive and negative values as long as Proposition 2 is satisfied to guarantee the existence of a unique solution that is periodic. Assume that the set-valued functions $\nu \mapsto \{\gamma | f_1(\nu, \gamma) = 0\}$ and $\nu \mapsto \{\gamma | f_2(\nu, \gamma) = 0\}$ admit explicit solutions given by

$$\gamma = g_1(\nu) \quad \text{and} \quad \gamma = g_2(\nu)$$

where $g_1, g_2 \in \mathscr{C}_a$ such that

$$f_1(\nu, g_1(\nu)) = 0 \quad \text{and} \quad f_2(\nu, g_2(\nu)) = 0.$$

To summarize, the curves described by g_1 and g_2 are the zero level set of the functions f_1 and f_2, which is the level set at height equal to zero.

By Proposition 2, both g_1 and g_2 split the input-output plane into two regions such that

$f_1(\nu, \gamma) < 0$	whenever	$\gamma > g_1(\nu)$;
$f_1(\nu, \gamma) > 0$	whenever	$\gamma < g_1(\nu)$;
$f_2(\nu, \gamma) > 0$	whenever	$\gamma > g_2(\nu)$;
$f_2(\nu, \gamma) < 0$	whenever	$\gamma < g_2(\nu)$.

Before being able to show that the butterfly hysteresis loop exists, the first two lemmas have to be taken into account.

Lemma 5. *If* $0 \leqslant \frac{dg_1(\nu)}{d\nu} \leqslant L_1$ *for all* $\nu \geqslant u_+(0)$ *then for all* $\gamma_0 \leqslant g_1(u_+(0)), y_{u_+}(\nu, \gamma_0) \leqslant g_1(\nu)$ *for all* $\nu \geqslant u_+(0)$.
The other way around, if $-L_2 \leqslant \frac{dg_2(\nu)}{d\nu} \leqslant 0$ *for all* $\nu \leqslant u_-(0)$ *then for all* $\gamma_0 \leqslant g_2(u_-(0)), y_{u_-}(\nu, \gamma_0) \leqslant g_2(\nu)$ *for all* $\nu \leqslant u_-(0)$.

In this case L_1 and L_2 are the bounds for the derivatives.This lemma shows the positive invariance of the region below the curves g_1 and g_2 concerning their solutions y_{u_+} and y_{u_-}. Lemma 6 shows that the extended solutions y_{u_+} and y_{u_-} in the reverse direction intersect with the zero level set curve g_2 and g_1, respectively, under certain assumptions.

Lemma 6. *Assume that the assumptions in Lemma 5 hold. Furthermore, assume that*

$$(f_1(\nu_1, \gamma) - f_1(\nu_2, \gamma))(\nu_1 - \nu_2) < 0,$$

for every $\nu_1, \nu_2, \gamma \in \mathbb{R}$ *and let* $\nu_a, \gamma_a \in \mathbb{R}$ *be such that* $\gamma_a = g_1(\nu_a) < g_2(\nu_a)$. *Then there exists* $\nu_b < \nu_a$ *such that* $y_{u_+}(\nu_b, \gamma_a) = g_2(\nu_b)$.
Likewise, suppose that

$$(f_2(\nu_1, \gamma) - f_2(\nu_2, \gamma))(\nu_1 - \nu_2) > 0,$$

for every $\nu_1, \nu_2, \gamma \in \mathbb{R}$ *and let* $\nu_a, \gamma_a \in \mathbb{R}$ *be such that* $\gamma_a = g_1(\nu_a) > g_2(\nu_a)$. *Then there exists* $\nu_b > \nu_a$ *such that* $y_{u_-}(\nu_b, \gamma_a) = g_1(\nu_b)$.

When one combines Lemma 5 and Lemma 6, it can be shown that the extended solutions in the negative direction of the input corresponding to the solutions intersect with the level set functions g_1 and g_2 can either have positive and negative slopes respectively or have negative and positive slopes respectively. The proof of 5 and Lemma 6 can be found in [3]. The main result can be found in 3. This proposition states existence of y_{u_-} and y_{u_+} with intersections.

Proposition 3. *Assume that the hypothesis in both Lemma 5 and Lemma 6 are satisfied. Let $\nu_f \in \mathbb{R}$ be such that $g_1(\nu_f) = g_2(\nu_f)$. Then $\forall\ \nu_{a_+} < \nu_f$, there exists $nu_{\min}, \nu_x, \nu_{a_-}$ and $\nu_{\max} \in \mathbb{R}$ such that $nu_{\min} < \nu_x < \nu_{a_-} < \nu_{\max}$ and*

$$\begin{aligned}
y_{u_+}(\nu_{\min}, g_1(\nu_{a_+})) &= y_{u_-}(\nu_{\min}, g_2(\nu_{a_-})) \\
y_{u_+}(\nu_x, g_1(\nu_x)) &= y_{u_-}(\nu_x, g_2(\nu_{a_-})) \\
y_{u_+}(\nu_{\max}, g_1(\nu_{a_+})) &= y_{u_-}(\nu_{\max}, g_2(\nu_{a_-}))
\end{aligned}$$

This proposition shows that the solutions $y_{u_+}(\cdot, g_1(\nu_{a_+}))$ and $y_{u_-}(\cdot, g_2(\nu_{a_-}))$ which respectively intersect g_1 and g_2 at ν_{a_+} and ν_{a_-} also intersect each other at $\nu_{\min}, \nu_x$ and $\nu_{\max}$. The proof of Proposition 3 can be found in [3].

Proposition 3 is a very strong proposition. When one combines Proposition 2, which shows the accommodation property, and Proposition 3, which is on the existence of an invariant butterfly loop, the conclusion is that the input-output phase plot will converge to the butterfly hysteresis loop for every initial value $\gamma_0 \in \mathbb{R}$ of the output when one applies a periodic input $u_p \in \mathscr{C}_a(\mathbb{R}_+, \mathbb{R})$ with only one maximum given by $\nu_{\max}$ and one minimum $\nu_{\min}$ on the periodic interval. Let us next consider an example of the Duhem butterfly hysteresis operator.

Example 4. First define the curves g_1 and g_2 satisfying the conditions of Lemma 5.

$$\begin{aligned}
g_1(\nu, \gamma) &:= a_1 + a_2\nu + a_3\nu^3, \\
g_2(\nu, \gamma) &:= -b_1 + b_2\nu - b_3\nu^3.
\end{aligned}$$

Next define f_1 and f_2 such that the curves corresponds to the zero level set, meaning that $f_1(\nu, f_1(\nu)) = f_2(\nu, f_2(\nu)) = 0$, and both assumptions in Lemma 6 and Proposition 3 are satisfied. Accordingly, f_1 and f_2 are chosen to be

$$\begin{aligned}
f_1(\nu, \gamma) &:= (a_1 + a_2\nu + a_3\nu^3 - \gamma), \\
f_2(\nu, \gamma) &:= (b_1 + b_2\nu + b_3\nu^3 + \gamma).
\end{aligned}$$

To show whether there exists a unique pair such that the generated sequences are convergent, one has to check the inequalities in Proposition 2. One needs to have that

$$\begin{aligned}(f_1(\nu,\gamma_1) - f_1(\nu,\gamma_2))(\gamma_1 - \gamma_2) < 0,\\ (f_2(\nu,\gamma_1) - f_2(\nu,\gamma_2))(\gamma_1 - \gamma_2) > 0,\end{aligned}$$

By filling in the chosen f_1 and f_2,

$$\begin{aligned}-(\gamma_1 - \gamma_2)^2 < 0,\\ (\gamma_1 - \gamma_2)^2 > 0,\end{aligned}$$

which is trivially true. Assume for simplicity that $a_1 = a_3 = b_1 = b_3 = 1$ and $a_2 = b_2 = 2$, then

$$\begin{aligned}f_1(\nu,\gamma) &:= (1 + 2\nu + \nu^3 - \gamma),\\ f_2(\nu,\gamma) &:= (1 + 2\nu + \nu^3 + \gamma).\end{aligned}$$

This can be rewritten as the following ordinary differential equation

$$\dot{y}(t) = \dot{u}(t) + 2u(t)\dot{u}(t) + u(t)^3\dot{u}(t) - y(t)|\dot{u}(t)|.$$

Consider again a periodic input function $u(t) = 5\sin(t)$. Again, using MATLAB's built-in function ODE45, the ordinary differential equation can be solved. The hysteresis loop obtained from the Duhem butterfly hysteresis operator when a periodic input is applied can be found in Figure 5

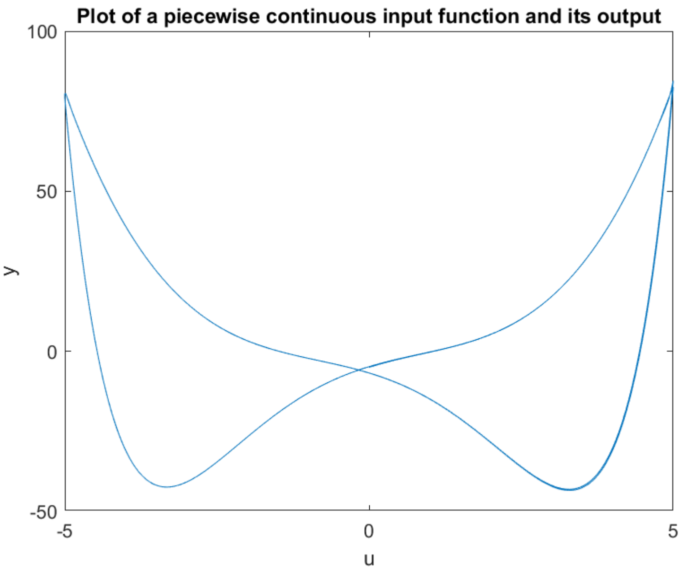


Figure 5: Duhem Butterfly Hysteresis Operator

3 Identification

The goal of system identification is to find mathematical models of dynamical systems from observed input-output data[16]. In this section, two ways to identify a mathematical model will be studied. First of all, the classical approach will be taken. In the classical approach, the focus is on prediction error methods. However, prediction error methods have some disadvantages. An example of a disadvantage is overfitting. After studying these disadvantages, another approach that does not have those disadvantages will be introduced. In particular, we will discuss kernel methods that have been popularized in the machine learning community.

3.1 Classical Methods

The most used approach to identify dynamical systems is the parametric prediction error method. Those classical methods in which mathematical statistics play an important role are widely studied [6]. In this method, a data set given by $\mathcal{D}$ is considered:

$$\mathcal{D} = \{u(1), y(1), \dots, u(N), y(N)\},$$

where $u(i) \in \mathbb{R}^m$ and $y(i) \in \mathbb{R}^p$ for $i = 1, \dots, N$. The model predicted values given by $\hat{y}(t|\theta)$ are compared to the actual output in the following function which represents the squared error of the outputs

$$V_N(\theta) = \sum_{t=1}^{N} (y(t) - \hat{y}(t|\theta))^2.$$

In order to find the optimal parameter vector θ, one would like to minimize $V_N(\theta)$ and hence the mathematical problem is given by

$$\min_{\theta} V_N(\theta) = \sum_{t=1}^{N} (y(t) - \hat{y}(t|\theta))^2.$$

The predictor output $\hat{y}(t|\theta)$ can be found by describing the model as a predictor of the next output and parametrizing the predictor using a finite-dimensional parameter vector, which leads to

$$\hat{y}(t|\theta) = f(\mathcal{D}^{t-1}, \theta),$$

where $\mathcal{D}^{t-1}$ is the data set at the previous time. An example of a predictor of the output is the predictor of the linear difference equation.

Example 5. The linear difference equation is given by

$$y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = b_1 u(t-1) + \cdots + b_m u(t-m).$$

The predictor output is given as

$$\begin{aligned}\hat{y}(t|\theta) &= f(\mathcal{D}^{t-1}, \theta) \\ &= a_1 y(t-1) + \cdots + a_n y(t-n) - b_1 u(t-1) - \cdots - b_m u(t-m),\end{aligned}$$

where $\mathcal{D}^{t-1}$ is the data set up to time $t-1$. After parameterizing the predictor, the model-predicted values are given by

$$f(\mathcal{D}^{t-1}, \theta) = \theta^T \varphi(t),$$

with the following parameters

$$\begin{aligned}\theta &= \begin{bmatrix} a_1 & \dots & a_n & b_1 & \dots & b_m \end{bmatrix}^T \\ \varphi(t) &= \begin{bmatrix} -y(t-1) & \dots & -y(t-n) & u(t-1) & \dots & u(t-m) \end{bmatrix}^T,\end{aligned}$$

To minimize the error given by the difference in the model predicted values and the actual output, a decent parameter estimate has to be found. The parameter estimate is given by $\hat{\theta}_N = \arg\min_\theta V_N(\theta)$, where $\arg\min_\theta$ returns the input for minimum output over all θ. The parameter estimate can be rewritten in the following way.

$$\begin{aligned}\hat{\theta}_N &= \arg\min_\theta V_N(\theta) \\ &= \arg\min_\theta \sum_{t=1}^{N} (y(t) - \hat{y}(t|\theta))^2 \\ &= \arg\min_\theta \sum_{t=1}^{N} (y(t) - \theta^T \varphi(t))^2.\end{aligned}$$

In order to find the solution, one needs to minimize the cost function

$$\min_\theta ||A^T \theta - y||^2,$$

where A is the matrix given by

$$A = \begin{bmatrix} \varphi(n)^T \\ \vdots \\ \varphi(N)^T \end{bmatrix},$$

if $n \geqslant m$ and the output y is given by the vector

$$y = \begin{bmatrix} y(n)^T \\ \vdots \\ y(N)^T \end{bmatrix}.$$

The solution to the least squares problem is given by

$$\hat{\theta} = A^{\dagger} y,$$

where $A^{\dagger}$ represents the pseudo-inverse of the matrix A [17].

In the above example, the predictor depends linearly on the parameters, and θ could be found using standard least squares. In general, however, the predictor may depend on θ in a nonlinear way which leads to more complex optimization problems. Furthermore, the number of predictors is large in many physical problems. With this higher dimensionality, there are more unknowns to estimate and thus the estimation of the parameter is more complex.

In physical problems, often noise is introduced in a system. A noise source describes the perturbations of a system due to for example vibrations and measurement inaccuracies. Note that if the noise source follows a Gaussian distribution, the method described above coincides with the well-known Maximum Likelihood method[10]. This method is widely used because it can be applied to a wide spectrum of model parametrizations to which it gives very good asymptotic properties. Note that the prediction error method has some advantages as it can be used in a closed-loop system and it has good asymptotic properties for many parametrizations [1]. However, a lot of disadvantages occur due to the nonlinearity and the high dimensionality. Two examples are overfitting and the bias-variance dilemma.

3.2 Overfitting

Overfitting occurs when the provisional model tries to account for every possible trend or structure in the training set. When a model has to be found, a certain balance between the model's complexity and its generalizability to the test and training sets has to be established. With a highly complex model, the accuracy is increased but a less general model is obtained. This is called overfitting. The question that arises is the question when a model is complex enough to describe the data but is also generalizable enough to not take every possible structure in the dataset into account. An example of polynomials with different degrees can be found in Figure 6 [4].

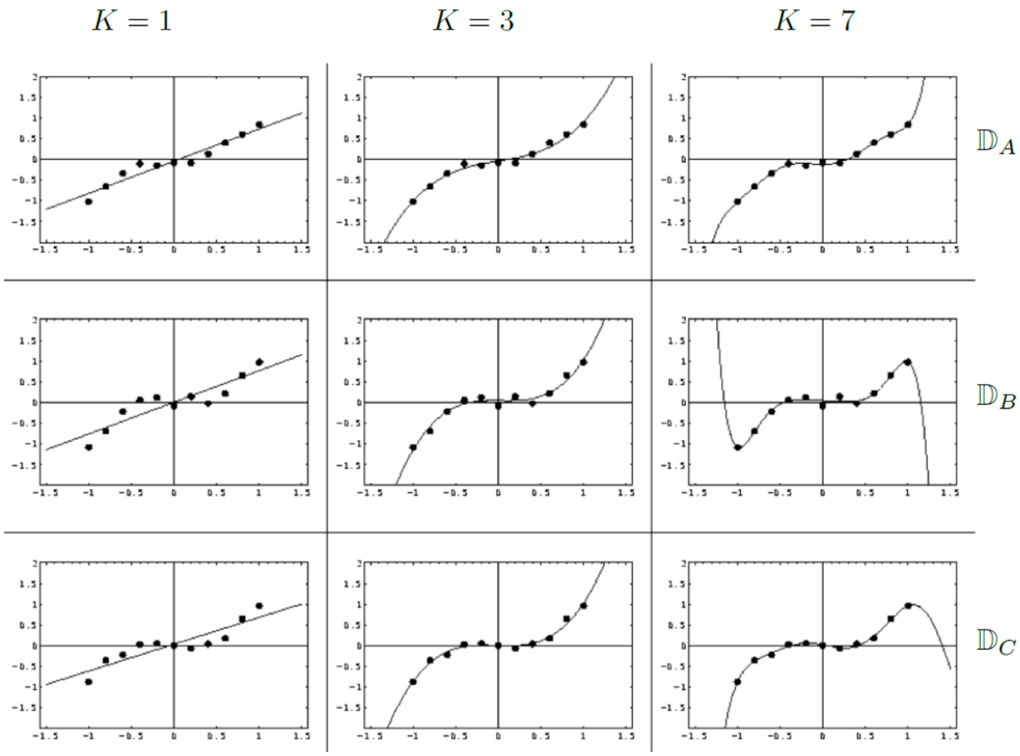


Figure 6: The concept of overfitting illustrated.

When the data set described by $\mathcal{D}$ is considered, it can be seen that a third-order polynomial fit might be a sufficient choice. However, there is always a higher-order polynomial for which the error is smaller.

It has to be decided when the polynomial is a sufficient approximation since higher order polynomials are more complex and computationally more costly. Choosing the right model class depends a lot on the data set $\mathcal{D}$ that is investigated and is a complex task [9], [12].

3.3 Bias-Variance Dilemma

Furthermore, the bias-variance dilemma occurs. Bias represents the deviation of the mean prediction from the true target. The average is taken over many randomized data sets. When the bias is small, there is a small error between the hypotheses and the unknown target rule. Variance measures how much individual predictions differ from the mean prediction. When the variance is small, the outcome has robust behavior. In Figure 7, it can be seen that for the data points of model A, the white dots, the bias is small and the variance is large [4]. If one takes the average of the predictions, this is close to the true target. However, the points are located all over the figure. The points are not close to the target, this means that the variance is large. On the other hand, if one looks at the fits of model B, the green dots, the bias is large and the variance is small. The fits of model B are not located near the target and thus the bias is large. The green dots are located close to the average of all model fits and hence there is a robust system meaning that the variance is small.

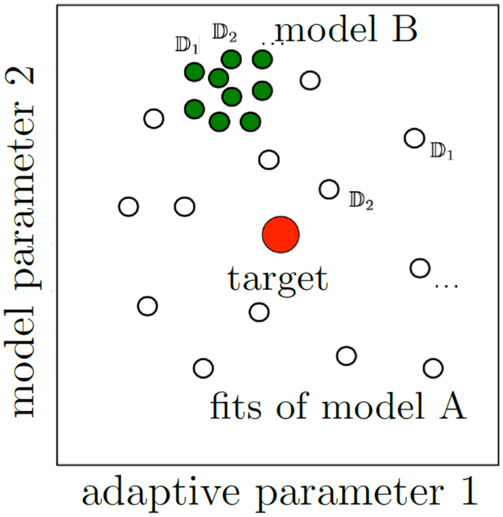


Figure 7: The bias-variance dilemma illustrated.

One would like to have a robust and faithful approximation of the system. This means that both the bias and the variance must be small. However, this is often a conflicting aim. In practice, one cannot have both a small variance and a small bias. Therefore, a conflict arises. A decision on which higher values for the bias and variance has to be chosen. This is dependent on the mathematical model that is studied and is called the bias-variance dilemma [4].

From these two disadvantages, it can be seen that an issue is to choose a suitable model order. One has to decide when the data is well represented but the model is not overfitted and thus overly complex. An overly complex model might lead to incorrectly predicted output when new data is put into the system. Furthermore, the problem of minimizing an overfitted model may lead to an ill-posed problem. The solution may become highly sensitive to small perturbations of the data y_i. This means that the model has a higher variance and thus the obtained model is less robust. An identification method with lower flexibility of the model class which can keep the well-posedness of the solution is desired. The identification method that satisfies these favorable properties is the function estimation by regularized kernel methods [16].

3.4 Kernel Methods

Consider the function $g : \mathscr{X} \to \mathbb{R}$. For kernel methods, instead of constraining the unknown function g to specific parameters, g is searched over a function space $\mathscr{H}$ which is possibly infinite-dimensional. To avoid overfitting and ill-posedness, a regularization term $J(g)$ is added to penalize undesired behaviours. Hence the objective functional becomes

$$\min_{g\in\mathscr{H}} \left(\sum_{i=1}^{N} (y_i - g(x_i))^2 + \gamma J(g) \right), \tag{3}$$

where γ is a positive constant. When $\mathscr{H}$ is a Hilbert space, mathematical analysis of regularization methods is possible [16]. Recall that a Hilbert space is a complete inner product space. This is a space with inner product $\langle \cdot, \cdot \rangle$ and since it is complete, every Cauchy sequence converges with respect to the induced norm $||g||_{\mathscr{H}} = \sqrt{(g, g)_{\mathscr{H}}}$ [8].

In this case, a specific choice for the regularizer is $J(g) = ||g||^2_{\mathscr{H}}$. Hence, the objective functional is given by

$$\min_{g \in \mathscr{H}} \left(\sum_{i=1}^{N} (y_i - g(x_i))^2 + \gamma ||g||^2_{\mathscr{H}} \right).$$

A function space that is such that the problem is well-posed and hence there exists a unique solution that is not too sensitive to data perturbations is the reproducing kernel Hilbert space.

Definition 6. A Hilbert space of functions $g : \mathscr{X} \to \mathbb{R}$ is called a reproducing kernel Hilbert space if

$$\forall x \in \mathscr{X}, \quad \exists C_x < \infty \quad : \quad |g(x)| \leqslant C_x ||g||_{\mathscr{H}}, \quad \forall g \in \mathscr{H}$$

A reproducing kernel Hilbert space is abbreviated as RKHS. The concept of RKHS is linked to the concept of positive semidefinite kernels.

Definition 7. A function $K(x_i, x_j)$ is symmetric if $K(x_i, x_j) = K(x_j, x_i)$A symmetric function $K : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is a positive semidefinite kernel if for all $p \in \mathbb{N}$, it holds

$$\sum_{i=1}^{p} \sum_{j=1}^{p} a_i a_j K(x_i, x_j) \geqslant 0, \quad \forall (x_k, a_k) \in (\mathscr{X}, \mathbb{R}), \quad k = 1, \ldots, p.$$

An example of such a positive semidefinite kernel is the Gaussian Kernel.

Example 6. The Gaussian Kernel is given by

$$K(x_1, x_2) = \exp\left(-\frac{||x_1 - x_2||^2}{2\sigma^2} \right),$$

where x_1 and x_2 are vectors that represent the coordinates of the data points and σ is the standard deviation.

The connection between RKHS and positive semidefinite kernels can be described by the theorem of Moore and Aronszajn [16]:

Theorem 2. *Given a positive semidefinite kernel K, there exists a unique RKHS of real valued functions defined over $\mathscr{X}$ whose reproducing kernel is K. Conversely, to every RKHS $\mathscr{H}$ there corresponds a unique positive semidefinite kernel K, the reproducing kernel, such that the reproducing property holds:*

$$g(x) = \langle g, K_x \rangle_{\mathscr{H}}, \quad \forall (x, g) \in (\mathscr{X}, \mathscr{H}),$$

where the kernel section $K_x \in \mathscr{H}$ centered at x is defined as $K_x(a) := K(x, a)$, $\forall a \in \mathscr{X}$.

From this theorem, it follows that the Hilbert space is completely determined by its reproducing kernel and hence the kernel choice specifies both the function space $\mathscr{H}$ as the regularizer. Furthermore, for modeling, a single positive semidefinite kernel function that encodes the desired properties of the function can be chosen [16]. An example that can be used to show both the Moore-Aronszajn theorem and the Representer theorem is a space of absolutely continuous functions.

Example 7. Consider an absolutely continuous function $f : [0, 1] \to \mathbb{R}$. The function f is searched over a function space $\mathscr{H}$. The Hilbert space $\mathscr{H}$ is completely determined by its reproducing kernel and therefore first the RKHS has to be found. Assume that $\mathscr{H}$ is the set of all f such that f' is square-integrable and $f(0) = f(1) = 0$. In this example, it will be shown that $\mathscr{H}$ is an RKHS. Make $\mathscr{H}$ a Hilbert space by defining the following inner product

$$\langle f, g \rangle = \int_0^1 f'(t) g'(t) dt.$$

Since $0 \leqslant x \leqslant 1$ and $f \in \mathscr{H}$,

$$f(x) = \int_0^x f'(t) dt.$$

In order to be a RKHS, a C_x must be found such that

$$|f(x)| \leqslant C_x ||f||_{\mathscr{H}}.$$

In order to find C_x, first let $0 \leqslant x \leqslant 1$ and let $f \in \mathscr{H}$. Since f is assumed to be absolutely continuous,

$$\begin{aligned} f(x) &= \int_0^x f'(t) dt \\ &= \int_0^1 f'(t) \mathscr{X}_{[0,x]}(t) dt, \end{aligned}$$

where $\mathscr{X}_{[0,x]}$ is used to transform the integral upper boundary from x to 1. By the Cauchy-Schwartz inequality

$$\begin{aligned} |f(x)| &\leqslant \sqrt{\int_0^1 f'(t)^2 dt} \sqrt{\int_0^1 \mathscr{X}_{[0,x]}(t) dt} \\ &= \sqrt{x} ||f||_{\mathscr{H}}, \end{aligned}$$

and hence $C_x = \sqrt{x}$ [15]. Hence, the chosen set $\mathscr{H}$ is a RKHS. Next, one would like to find the reproducing kernel. Let $K(t,x) = k_x(t)$. For the function $f(x)$ a boundary value problem that has a continuous solution can not immediately be derived. However, it is known that $k_x(t)$ exists and is continuous, and hence $f(x)$ is rewritten, and integration by parts will be applied.

$$\begin{aligned} f(x) = \langle f, k_x \rangle &= \int_0^1 f'(t) k_x'(t) dt \\ &= -\int_0^1 f(t) k_x''(t) dt. \end{aligned}$$

Introduce the Dirac-Delta function $\delta_x = -k_x''$. The problem can be rewritten as

$$f(x) = \int_0^1 f(t) \delta_x(t) dt,$$

with $k_x(0) = k_x(1) = 0$. Note that this boundary value problem is the Green's function. Therefore, the solution to this problem is

$$K(t,x) = k_x(t) = \begin{cases} (1-x)t, & t \leqslant x, \\ (1-t)x, & t \geqslant x. \end{cases}$$

And hence the reproducing kernel that is used to find the solution to the minimization problem is found [15].

The solution to the optimization problem, given by Equation 3, is a finite-dimensional representation. This can be captured in the representer theorem and the representer kernel is used [16].

Theorem 3. *If $\mathscr{H}$ is a RKHS, the minimizer of* $\min_{g\in\mathscr{H}} \left(\sum_{i=1}^{N}(y_i - g(x_i))^2 + \gamma J(g)\right)$, *is given by*

$$\hat{g}(x) = \sum_{i=1}^{N} \hat{c}_i K_{x_i}(x),$$

with $\hat{c} = \left[\hat{c}_1, \ldots, \hat{c}_N\right]^T$ *is given by* $\hat{c} = (\mathbf{K} + \gamma I_N)^{-1} Y$, *where* $\mathbf{K}$ *is a Gram matrix of size* $N \times N$, I_N *the identity matrix of size* $N \times N$ *and* $Y = (y_1, \ldots, y_N)$ *is the column vector containing all available output measurements.*

The Gram matrix is the matrix K with elements $K_{ij} = k(x_i, x_j)$. If $k(x_i, x_j) = x_i^T x_j$, the Gram matrix can be expressed in terms of the matrix $X = \begin{bmatrix} x_1 & \ldots & x_m \end{bmatrix}$ as

$$G = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \begin{bmatrix} x_1 & \ldots & x_N \end{bmatrix}$$

If the kernel is symmetric and positive semidefinite, the Gram matrix is also symmetric and positive semi-definite [18]. The reproducing kernel plays an important role and therefore it must be chosen carefully. A large variety of positive semidefinite kernel functions have been studied. The representer theorem is an important result as an optimization over an infinitely large space is possible.

Note that from the solution of the variational problem

$$\min_{g\in\mathscr{H}} \left(\sum_{i=1}^{N}(y_i - g(x_i))^2 + \gamma ||g||_{\mathscr{H}}^2\right),$$

a finite-dimensional representation is found. Just as in the classical approach, the optimal function is a linear combination of basis functions. The number of basis functions is equal to the number of data pairs. The difference between the prediction error method and the kernel method is that the number of basis functions and data pairs are not fixed a priori [16].

3.5 Impulse Response Estimation Problem

In this section, kernel methods to the problem of continuous-time system identification will be applied via impulse response estimation. When a linear time-invariant system is excited by an impulse input and noisy samples of the output are collected, the linear system identification problem can be solved using the function estimation techniques previously described. A continuous-time single-input-single-output dynamic system is considered. Assume that this system is linear, time-invariant, and causal. The output error model is given by

$$y(t_i) = \int_0^\infty u(t_i - s)g(s)ds + e_i \quad i = 1, \ldots, N.$$

Note that $y(t_i)$ denotes the output measured at time t_i and u the input. Furthermore, e_i is the white noise and the unknown function g is the system impulse response. To identify the system, the function $g : \mathbb{R}^+ \to \mathbb{R}$ must be reconstructed from an inputvector u and an output vector y. The impulse response will be estimated by an element of the RKHS which is associated with the kernel $K : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}$. The objective functional is given as

$$\min_{g \in \mathscr{H}} \left(\sum_{i=1}^{N} (y_i - L_i[g])^2 + \gamma ||g||^2_{\mathscr{H}} \right), \tag{4}$$

where $||g||^2_{\mathscr{H}}$ is the regularizer. In this case, $g(x_i)$ is replaced by $L_i[g]$ since the data of the nature is more complex. The functional $L_i[g]$ is given by

$$L_i[g] = \int_0^\infty u(t_i - s)g(s)ds.$$

The question that arises is, can one find from the solution of the variational problem a finite-dimensional representation? The answer is positive if the linear functionals $L_i[g]$ are continuous on $\mathscr{H}$, i.e.

$$\forall i, \quad \exists C_i < \infty \quad : \quad |L_i[g]| \leqslant C_i ||g||_{\mathscr{H}}, \quad \forall g \in \mathscr{H}.$$

Before stating the main result, the output kernel O must be defined. The output kernel is given by

$$O(t, \tau) = \int_0^\infty u(t - x) \left(\int_0^\infty u(\tau - a)K(x, a)da \right) dx.$$

The output kernel matrix $\mathbf{O} \in \mathbb{R}^{N \times N}$ is a positive semidefinite matrix with (i, j) entry given as

$$\mathbf{O}_{ij} = L_i[L_j[K]] = O(t_i, t_j).$$

The main result is the representer theorem for system identification [16].

Theorem 4. *If $\mathscr{H}$ is a RKHS induced by K and each $L_i : \mathscr{H} \to \mathbb{R}$ is a continuous linear functional, the minimizer is*

$$\hat{g}(x) = \sum_{i=1}^{N} \hat{c}_i L_i[K_x],$$

with $\hat{c} = [\hat{c}_1, \ldots, \hat{c}_N]^T$ and $\hat{c} = (\mathbf{O} + \gamma I_N)^{-1} Y$, where I_N represents the identity matrix of size N.

Thus, the solution to the optimization problem as given in Equation 4, admits a finite-dimensional representation. The choice of the kernel is an important decision to make as different kernels give different behaviour. Three examples and their behaviour can be found in Figure 8.

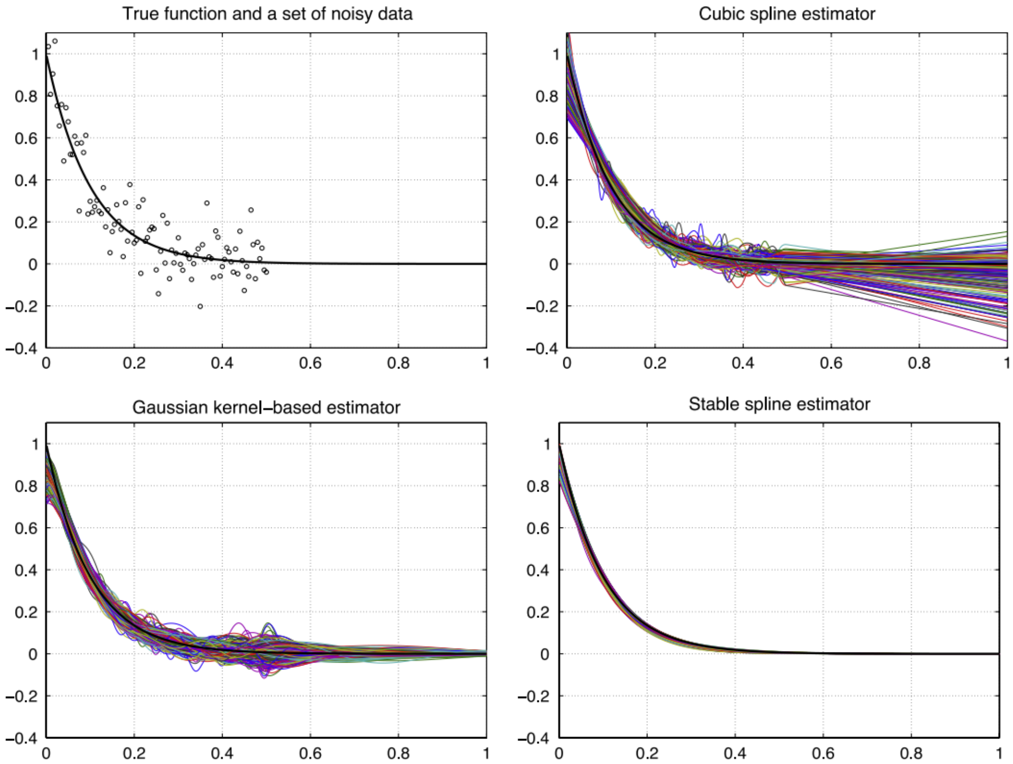


Figure 8: The different kernels used to obtain the impulse response estimates.

In this case, a stable linear system is considered and one would like to capture this linear system in the kernel. As mentioned previously, the choice of this kernel is very important. There are three examples written in Figure 8. In the left top figure, the true impulse response and a noisy data set can be seen. In the other three figures the true impulse response together with the impulse response estimates obtained using the Gaussian kernel, cubic spline kernel, and the stable spline kernel [16]. From this example, different behaviour in the sense of oscillations, convergence, and stability can be seen. One can conclude that for this linear, stable system the stable spline estimator is the most suitable one[16].

4 Hysteretic Deformable Mirror

The hysteretic deformable mirror is a deformable mirror with hysteresis behaviour and is abbreviated as HDM. The HDM is a new approach to the design of deformable mirrors. The goal of a deformable mirror is to correct the wavefront in advanced imaging systems. In Figure 9 the design of a HDM can be seen.

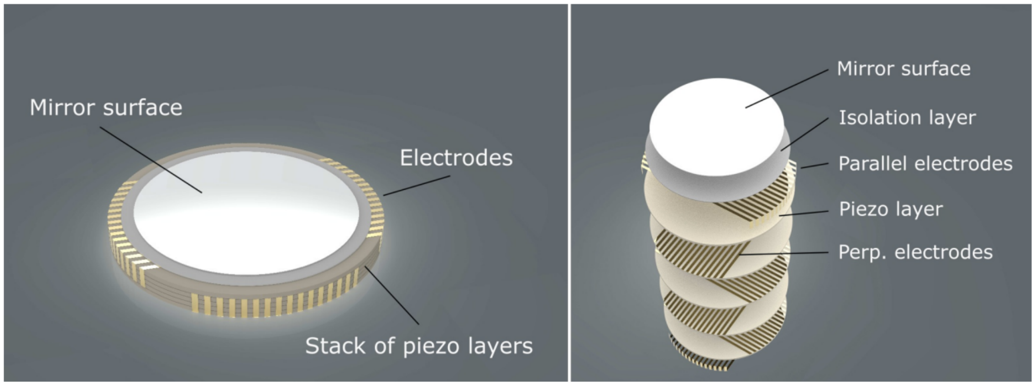


Figure 9: The conceptual design of the HDM together with a zoomed-in view of the HDM.

The HDM consists of a reflective top layer, thin piezo layers, and electrodes. The stack of thin piezo layers is separated from each other by top and bottom electrodes which consists of arrays of parallel metallic strips. The strips of the bottom and top electrodes are placed perpendicular to each other. Furthermore, the strips are equally aligned for all layers in the way that the top electrode strips of the layers lay directly above each other and are electrically connected. The same holds for the bottom layers. The electric field that is generated when a voltage over any two perpendicular strips is applied, causes a local deformation of the piezo layer that is sandwiched between the two perpendicular strips at the crossing of the electrodes. When this piezo layer deforms, the reflective top layer deforms as well and hence the mirror is deformable [7].

4.1 Experiments

The experiments have been executed together with SRON and ZIAM. Different pallets were given as samples. The samples are fabricated with Nb-doped PZT$-$ZrO$_2$. To conduct the voltage that will be applied, the samples had to be painted with silver paste. Thereafter, each sample was measured. The information on the diameter and thickness was needed in the program on the computer that performed the experiment.

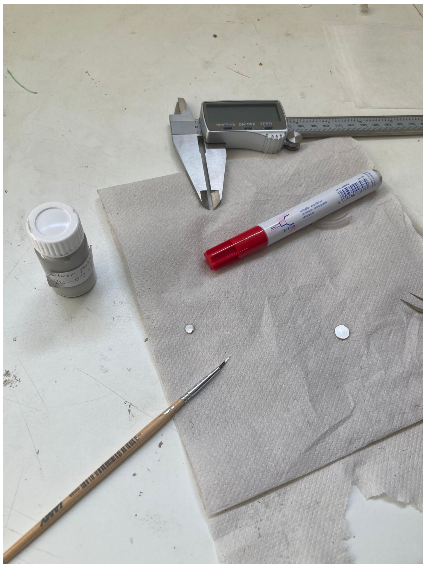


Figure 10: Silverpaste.

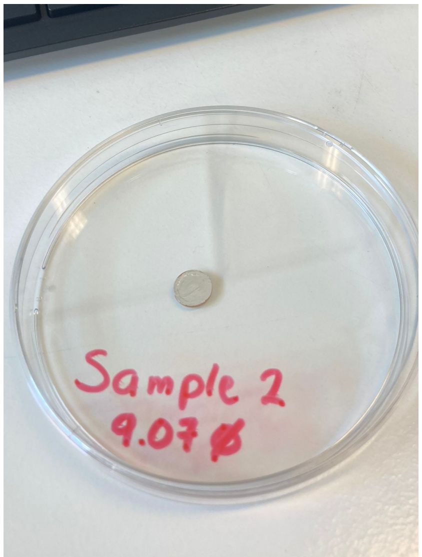


Figure 11: Sample.

Those samples were placed in a box where high voltages could be applied. Due to the sensitivity of piezo-electric material, before applying certain voltages and frequencies, a literature study has been performed. It can be found that from 600 V a switch of the dipoles can be seen. During the experiments, the voltages has been increased up to 1.3 kV. The frequency was 1 Hz during the whole experiment [7]. To have the most accurate result, the sample had to be placed as centered as possible.

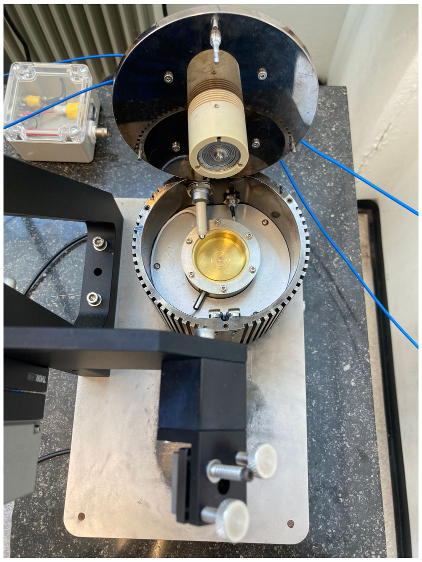


Figure 12: A Sample in the Box.

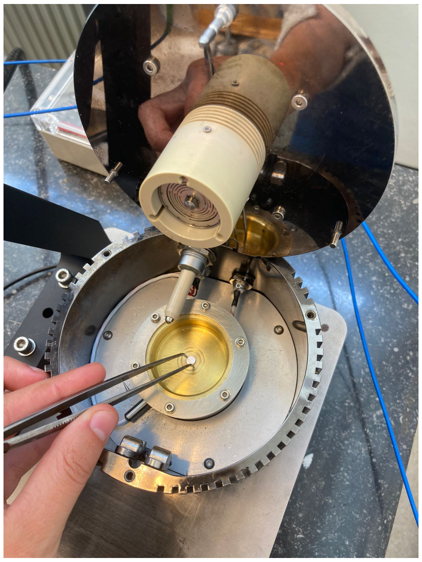


Figure 13: Placing a Sample.

After the closure of the box, the laser has to be calibrated. One would like to have a circle as in Figure 15 that has at least 70% to be accurate enough. This is an application of opto-mechatronics.

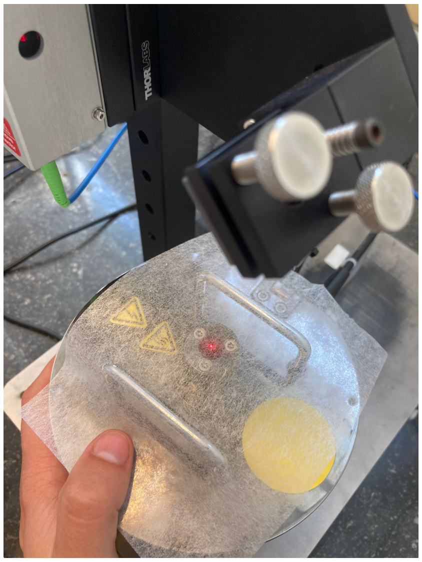
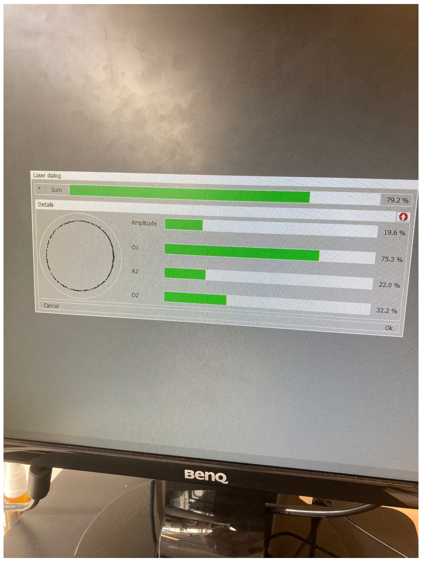


Figure 14: The laser.



Laser dialog

Sum 79.2 %

Details

Amplitude 19.6 %

O1 75.3 %

A2 22.0 %

O2 32.2 %

Figure 15: Calibrating.

During the experiment, a periodic input is applied. This periodic input obtains its maximum at 1.3 kV and its minimum at -1.3 kV. This input has been applied for one second. The periodic input $u(t)$ can be described by the following function.

$$u(t) = \begin{cases} 5200t, & \text{if} \quad 0 \leqslant t \leqslant 0.25, \\ -5200t + 2600, & \text{if} \quad 0.25 \leqslant t \leqslant 0.75, \\ 5200t - 5200, & \text{if} \quad 0.75 \leqslant t \leqslant 1. \end{cases}$$

This function can be seen in Figure 16.

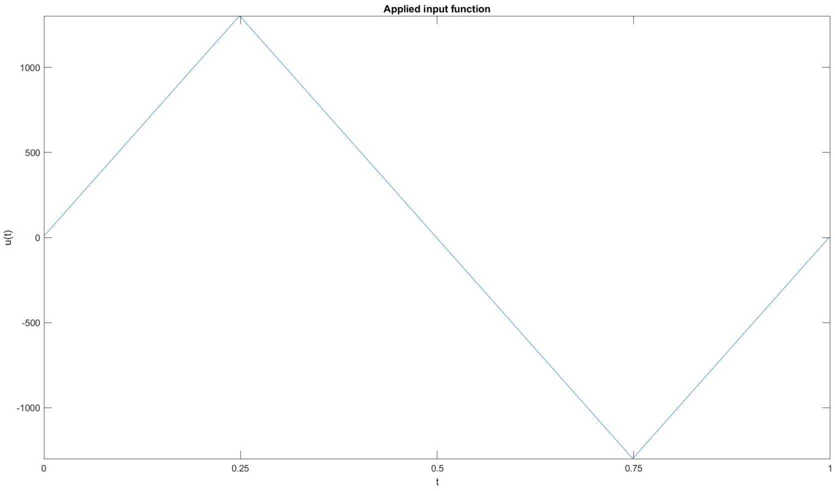


Figure 16: The input function of the HDM.

After applying the input, the output is obtained. This output can be found in Figure 17.

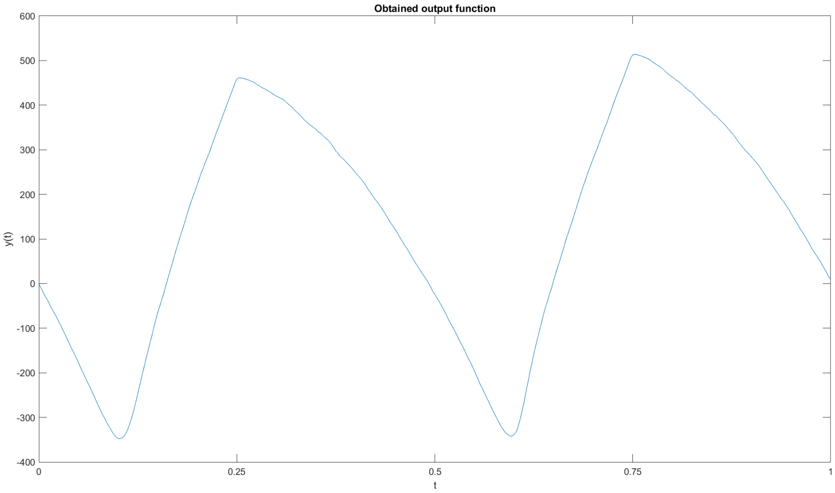


Figure 17: The output function of the HDM.

Combining both the input and the output, a butterfly-shaped loop is expected. The tested example indeed shows the behaviour as one can see in Figure 18.

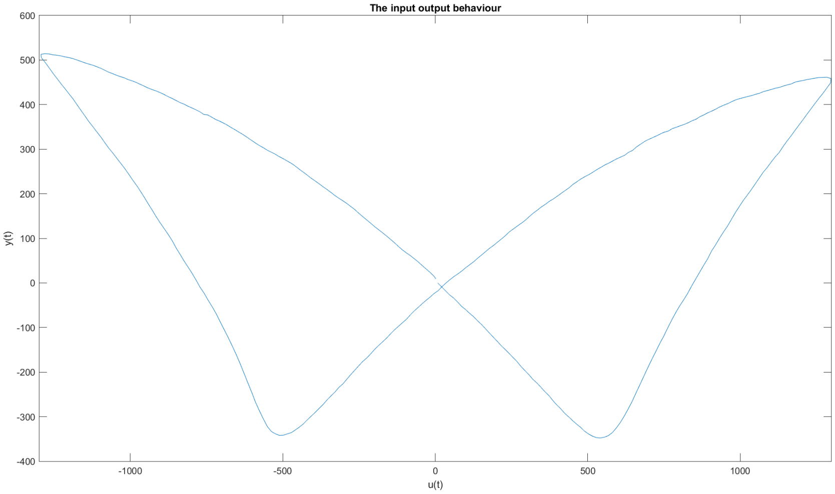


Figure 18: The input-output behaviour of the HDM.

The next step is to find the mathematical model of the dynamical system from the observed input-output data from the experiment with the HDM. The mathematical model can be described as an ordinary differential equation $\dot{y} = f(u, y)\dot{u}$.

4.2 Identification

Two ways of identification are used throughout this research. The prediction error method and the kernel method.

4.2.1 Prediction Error Method

The first way of identification is a classical method. It is called the prediction error method as discussed in Section 3.1. To identify the system, one wants to find the parameters $a_0, \dots, a_5$ and $c_0, \dots, c_5$ such that the ordinary differential equation describes the data as accurately as possible. The two curves $c_1(\nu, \gamma)$ and $c_2(\nu, \gamma)$ that are used are given by:

$$c_1(\nu, \gamma) = \sum_{n=0}^{5} a_n \nu^n \qquad c_2(\nu, \gamma) = -\sum_{n=0}^{5} c_n \nu^n.$$

The functions f_1 and f_2 are therefore given by

$$f_1(\nu, \gamma) = -\gamma + \sum_{n=0}^{5} a_n \nu^n \qquad f_2(\nu, \gamma) = \gamma + \sum_{n=0}^{5} c_n \nu^n.$$

[2]. This gives the following ordinary differential equation

$$\dot{y}(t) = \begin{cases} (a_0 + a_1 u(t) + a_2 u^2(t) + a_3 u^3(t) + a_4 u^4(t) + a_5 u^5(t) - y(t))\dot{u}(t), & \text{if} \quad \dot{u}(t) \geqslant 0, \\ (c_0 + c_1 u(t) + c_2 u^2(t) + c_3 u^3(t) + c_4 u^4(t) + c_5 u^5(t) + y(t))\dot{u}(t), & \text{if} \quad \dot{u}(t) < 0. \end{cases}$$

To find the parameters a_n and c_n for $n = 0, \dots, 5$, the least squares method has to be applied. The least squares method finds the minimum over all parameters for the system $A\theta = b$. Note that since the differential equation consists of two parts, one for $\dot{u}(t) \geqslant 0$ and one for $\dot{u}(t) < 0$, also the parameter fit will be applied two times.

When $\dot{u}(t) \geqslant 0$, the parameters that have to be found are $a_0, \dots, a_5$, therefore θ has to be represented as

$$\theta = \begin{bmatrix} a_0 \\ \vdots \\ a_5 \end{bmatrix}$$

Note that the ODE can be rewritten as

$$\begin{aligned} \dot{y}(t) &= \left(-y(t) + \sum_{n=0}^{5} a_n u^n(t) \right) \dot{u}(t), \\ \dot{y}(t) &= -y(t)\dot{u}(t) + \left(\sum_{n=0}^{5} a_n u^n(t) \right) \dot{u}(t), \\ \dot{y}(t) + y(t)\dot{u}(t) &= \left(\sum_{n=0}^{5} a_n u^n(t) \right) \dot{u}(t) \end{aligned}$$

Therefore, the vector b is given by

$$b = \begin{bmatrix} \dot{y}(0) + y(0)\dot{u}(0) \\ \dot{y}(1) + y(1)\dot{u}(1) \\ \vdots \\ \dot{y}(T) + y(T)\dot{u}(T) \end{bmatrix},$$

and the matrix A can be denoted as

$$A = \begin{bmatrix} \dot{u}(0) & u(0)\dot{u}(0) & \dots & u^5(0)\dot{u}(0) \\ \dot{u}(1) & u(1)\dot{u}(1) & \dots & u^5(1)\dot{u}(1) \\ \vdots & \vdots & \vdots & \vdots \\ \dot{u}(T) & u(T)\dot{u}(T) & \dots & u^5(T)\dot{u}(T) \end{bmatrix}.$$

The solution to this least squares problem can be found by multiplying the vector b by the pseudo-inverse of the matrix A and is given by

$$\theta = A^{\dagger} b,$$

where θ represents the parameters a_i.

When $\dot{u}(t) < 0$, the parameters that have to be identified are $c_0, \ldots, c_5$. Therefore, θ is represented as

$$\theta = \begin{bmatrix} c_0 \\ \vdots \\ c_5 \end{bmatrix}.$$

The differential equation can be written as

$$\dot{y}(t) - y(t)\dot{u}(t) = \left(\sum_{n=0}^{5} c_n u^n(t) \right) \dot{u}(t).$$

Therefore, the matrix b is represented as

$$b = \begin{bmatrix} \dot{y}(0) - y(0)\dot{u}(0) \\ \dot{y}(1) - y(1)\dot{u}(1) \\ \vdots \\ \dot{y}(T) - y(T)\dot{u}(T) \end{bmatrix},$$

and A is as before. The solution to this problem is again given by

$$\theta = A^{\dagger} b,$$

where θ represents in this case the parameters $c_0, \ldots, c_5$ [17].

As one can see from the differential equation, the maximum order of the input $u(t)$ is five. As the maximum input that is applied in the experiment is 1300 volts, scaling is needed. If one does not scale the input $u(t)$ and the output $y(t)$ it is hard to find the right parameters since for $u(t)^5$ values up to $3.7 \times 10^1 5$ appear. In this case, the input function applied is

$$u(t) = \begin{cases} 5.2t, & \text{if} \quad 0 \leqslant t \leqslant 0.25, \\ -5.2t + 2.6, & \text{if} \quad 0.25 \leqslant t \leqslant 0.75, \\ 5.2t - 5.2, & \text{if} \quad 0.75 \leqslant t \leqslant 1. \end{cases}$$

By applying the least squares method to fit the data, the vectors θ are found. When the derivative of the input $\dot{u}(t)$ is positive, the vector θ is given by

$$\theta = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \approx \begin{bmatrix} -0.763 \\ -0.699 \\ 2.651 \\ -2.756 \\ -1.072 \\ -1.377 \end{bmatrix}.$$

In the case that $\dot{u}(t)$ is negative, the vector θ is

$$\theta = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} \approx \begin{bmatrix} 0.735 \\ -0.299 \\ -2.769 \\ 2.138 \\ 1.173 \\ -1.138 \end{bmatrix}.$$

When the ordinary differential equation

$$\dot{y}(t) = \begin{cases} (a_0 + a_1 u(t) + a_2 u^2(t) + a_3 u^3(t) + a_4 u^4(t) + a_5 u^5(t) - y(t))\dot{u}(t), & \text{if} \quad \dot{u}(t) \geqslant 0, \\ (c_0 + c_1 u(t) + c_2 u^2(t) + c_3 u^3(t) + c_4 u^4(t) + c_5 u^5(t) + y(t))\dot{u}(t), & \text{if} \quad \dot{u}(t) < 0. \end{cases}$$

is solved by using the ODE45 built-in function of MATLAB, the following result is obtained.

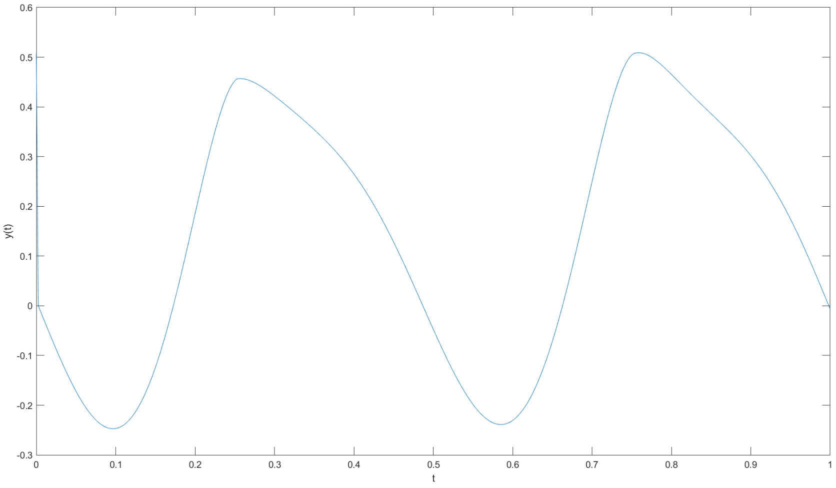


Figure 19: The output function by the Duhem Model.

When one compares the actual output with the obtained output by the approximation using the Duhem butterfly model, the following graph is obtained.

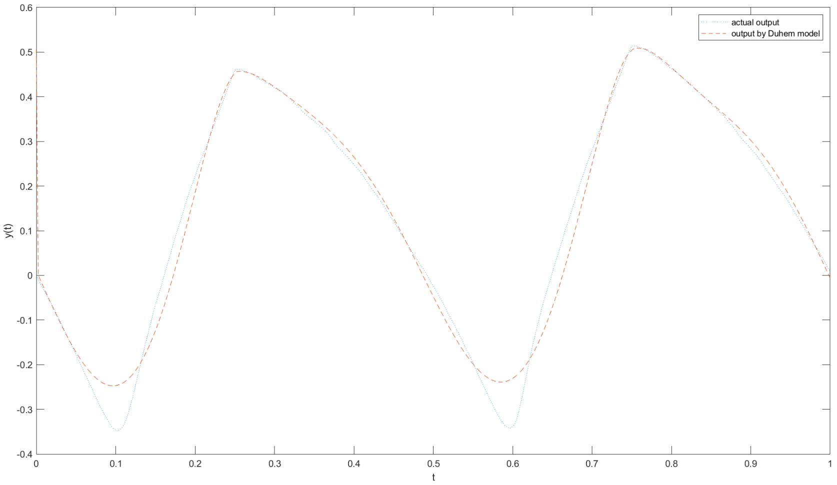


Figure 20: The actual output and approximated output compared.

When one compares the input-output behaviour of the data together with the approximation by the Duhem butterfly model, the following graph is obtained.

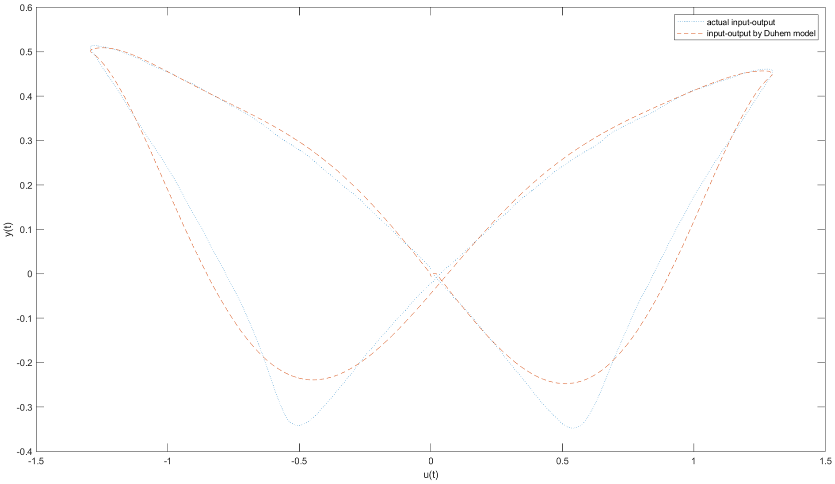


Figure 21: The actual input-output and approximated input-output behaviour compared.

The error can be computed by taking the norm of the vectors y and $\hat{y}$, where y represents the actual output of the system and $\hat{y}$ is the approximated output and hence,

$$\begin{aligned} \text{error} &= ||y - \hat{y}|| \\ &= 0.886. \end{aligned}$$

After deriving the prediction error method using the Duhem butterfly model with an error of 0.886, one would prefer to obtain a method that gives a smaller error.

4.2.2 Kernel Method

The second method that will be discussed is the kernel method. In this case, one looks for a function $\hat{f}(u)$ in the following manner

$$\begin{aligned}\dot{y} &= \dot{u} f(u, y)\\ \dot{y} &= \dot{u}\hat{f}(u) + y\dot{u}\\ \hat{f}(u) &= \frac{\dot{y}}{\dot{u}} - y \qquad (5)\end{aligned}$$

To identify $\hat{f}(u)$, one would like to find the f in a reproducing kernel Hilbert space. The least squares problem is in this case described by

$$\min_{f \in \mathcal{H}} \left(\sum_{i=1}^{n} ||y_i - f(u_i)||^2_{\mathcal{Y}} \right),$$

where u_i is the input at time t_i. The output y_i at time t_i is given by

$$y_i = \frac{\dot{y}(t_i)}{u(t_i)} - y(t_i),$$

since the to be identified function $\hat{f}(u)$ is described as in Equation 5. To identify f, one can directly use Theorem 3 [20]. Recall that Theorem 3 states that if $\mathscr{H}$ is an RKHS, the solution $\hat{g}(x)$ to the optimization problem is given by

$$\hat{g}(x) = \sum_{i=1}^{N} \hat{c}_i K_{x_i}(x),$$

with $\hat{c} = [\hat{c}_1, \ldots, \hat{c}_N]^T$ is given by $\hat{c} = (\mathbf{K} + \gamma I_N)^{-1} Y$, where $\mathbf{K}$ is a Gram matrix of size $N \times N$, I_N the identity matrix of size $N \times N$ and $Y = (y_1, \ldots, y_N)$ is the column vector containing all available output measurements. The Gram matrix $\mathbf{K}$ can be denoted in the following manner

$$\mathbf{K} = \begin{bmatrix} K(u_1, u_1) & K(u_1, u_2) & \ldots & K(u_1, u_n) \\ K(u_2, u_1) & K(u_2, u_2) & \ldots & K(u_2, u_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(u_n, u_1) & K(u_n, u_2) & \ldots & K(u_n, u_n) \end{bmatrix},$$

The system of linear equations can be rewritten as

$$(\mathbf{K} + \gamma I)(c_1, \ldots, c_n) = (y_1, \ldots, y_n).$$

To find the coefficients $(c_1, \ldots, c_n)$, the following system of linear equations has to be solved

$$(\mathbf{K} + \gamma I)(c_1, \ldots, c_n) = (y_1, \ldots, y_n),$$

which gives

$$(c_1, \ldots, c_n) = (\mathbf{K} + \gamma I)^{-1}(y_1, \ldots, y_n).$$

Note that to define the Gram matrix, a linear combination of suitable kernels has to be chosen. Those linear combinations can consist of different types of kernels such as the Laplacian kernel and the Gaussian kernel [20].

The first kernel that is chosen to identify the data from the hysteretic deformable mirror is the Gaussian kernel. The Gaussian kernel function is given by

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{\sigma^2}\right),$$

where σ is a parameter that can be adjusted to optimize the solution. In the MATLAB code in the appendix, it can be seen that first the Gram matrix $\mathbf{K}$ is defined. Thereafter, the coefficients c_j will be found. Thereafter the output $\hat{f}(u)$ of the kernel method is calculated. In order to compare this output with the actual output the ODE given by

$$\dot{y} = \dot{u}\hat{f}(u) + y\dot{u},$$

has to be computed. This can be done using the built-in function ODE45. In order to optimize the Gaussian kernel method, the parameter σ can be adjusted. To find the optimal value for σ, the error of different values of σ are considered. This error is calculated as follows:

$$\text{error} = ||y - \hat{\psi}||,$$

where y is the actual output by the system and $\hat{\psi}$ is the approximation of the output using the Gaussian kernel function. By trial and error, it is found that the optimal value for σ equals 0.1. This gives an error value of 0.355 which is significantly lower than the value of 0.886 which was found by using the Duhem butterfly model. This result can also be seen when the plots of the time, input, and output are generated. In Figure 22 the output for the time $t = 0$ to $t = 1$ second can be seen.

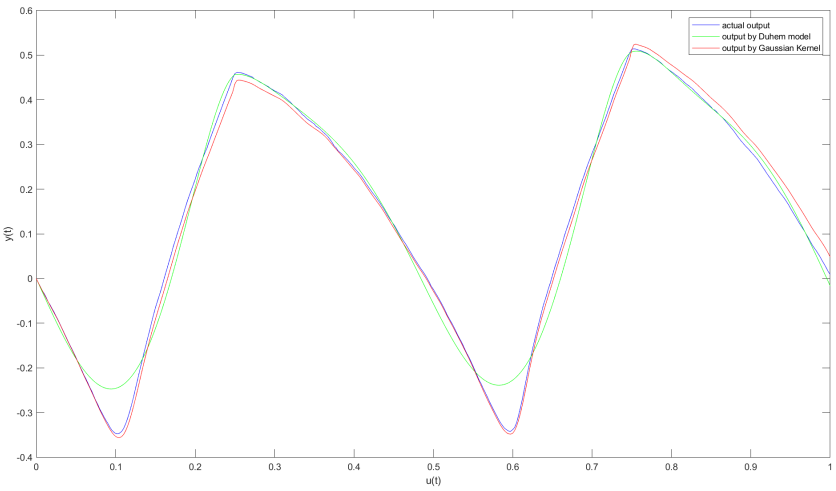


Figure 22: The output of the data, the output of the Duhem butterfly model, and the output of the Gaussian kernel method compared.

As one can see from this graph, especially in the angular parts in the lower part of the graph, the kernel method performs way better than the Duhem butterfly model. This part also explains the large difference in the error, the approximation of the Gaussian kernel method is more accurate. In Figure 23 the input-output behaviour is compared.

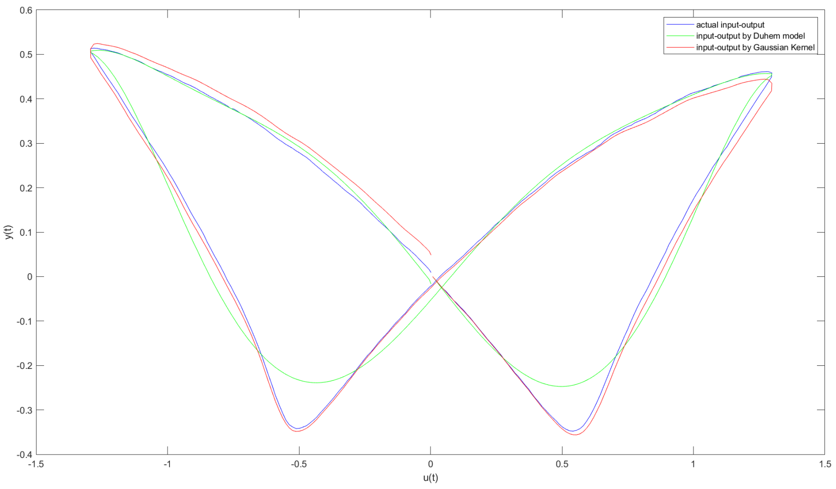


Figure 23: The input-output behaviour of the data, the Duhem butterfly model, and the Gaussian kernel method compared.

It can be seen that in the lower wings of the butterfly, the Gaussian kernel method performs better. The gap between the red plot and the blue plot is smaller than the gap between the blue and green plot. Thus, the error is approximately 2.5 times smaller which gives a significantly better result.

Another kernel function that can be chosen is the Laplacian kernel function. This kernel function is given by

$$k(x, y) = \rho \cdot \exp(-||x - y||)$$

By following a similar method as for the Gaussian kernel, the output of the Laplacian kernel function can be found. The optimal value for ρ is found by trial and error. The optimal value is $\rho = 5 \cdot 10^{-4}$ By using the solver ODE45 in MATLAB for which the code also can be found in the appendix, the following figures are obtained. In Figure 24 the output of the Laplacian kernel compared to the Duhem butterfly model and the actual output can be seen. One can conclude that this kernel method is better than the Duhem butterfly model just as the Gaussian kernel method.

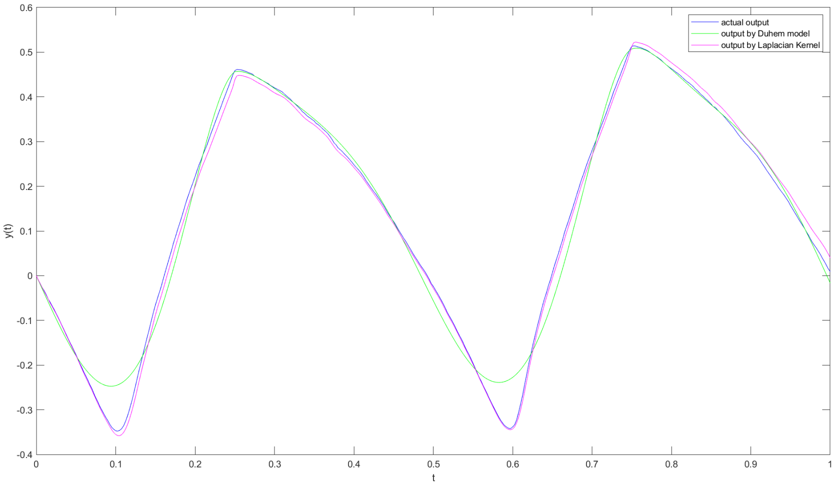


Figure 24: The output of the data, the output of the Duhem butterfly model, and the output of the Laplacian kernel method compared.

If one looks at Figure 25, it is not immediately clear that the approximations by the Laplacian kernel are better than the Gaussian kernel.

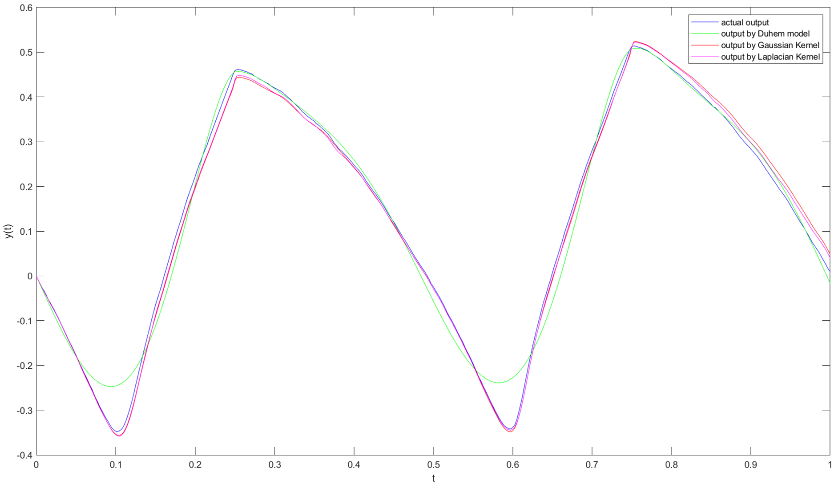


Figure 25: The output of the data, the output of the Duhem butterfly model the output of the Gaussian kernel method, and the output of the Laplacian kernel method compared.

However, when one calculates the error, the difference will be clear. In order to calculate the error, assume that y represents again the actual output of the system and $\hat{\xi}$ represents the approximation using the Laplacian kernel function. Then, the error is given by

$$\begin{aligned} \text{error} &= ||y - \hat{\xi}||, \\ &= 0.285. \end{aligned}$$

This error is smaller than the error of the Gaussian kernel method and this gives an error that is approximately three times smaller than the error of the Duhem butterfly model. In Figure 26 the input and output of the data, the Duhem butterfly model and both kernel functions can be seen.

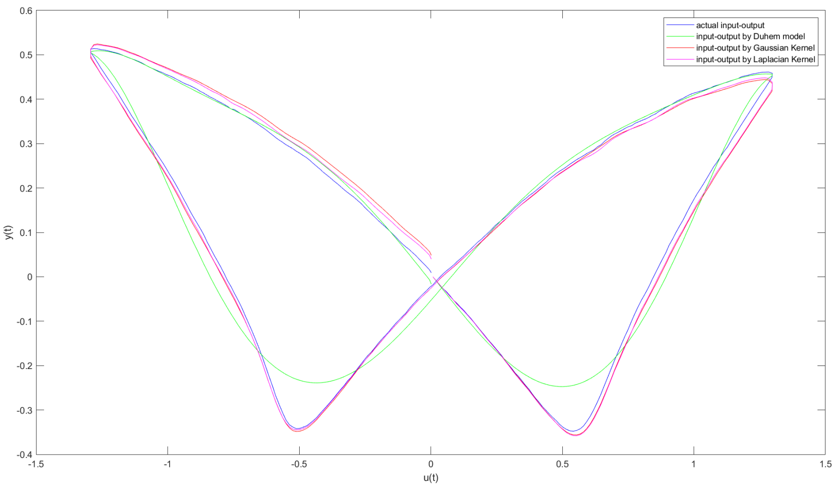


Figure 26: The input-output behaviour of the data, the Duhem butterfly model, the Gaussian kernel method, and the Laplacian kernel method compared.

Another way to choose the kernel is to choose a linear combination of multiple kernel functions. In this section, a combination of the Gaussian and Laplacian kernel function is chosen. In this case, the kernel function is given by

$$k(x, y) = \exp\left(-\frac{||x-y||^2}{\sigma^2}\right) + \rho \cdot \exp(-||x-y||)$$

By using a similar approach as for the Gaussian kernel and the Laplacian kernel, the output by the linear combination of the Gaussian and Laplacian kernel functionscan be obtained. Again, by trial and error, the optimal values for σ and ρ are found. The optimal values are $\sigma = 0.11$ and $\rho = 0.1$. In Figure 27 the output given by this kernel method can be seen. Again, this choice for the kernel gives a better result than the Duhem butterfly model.

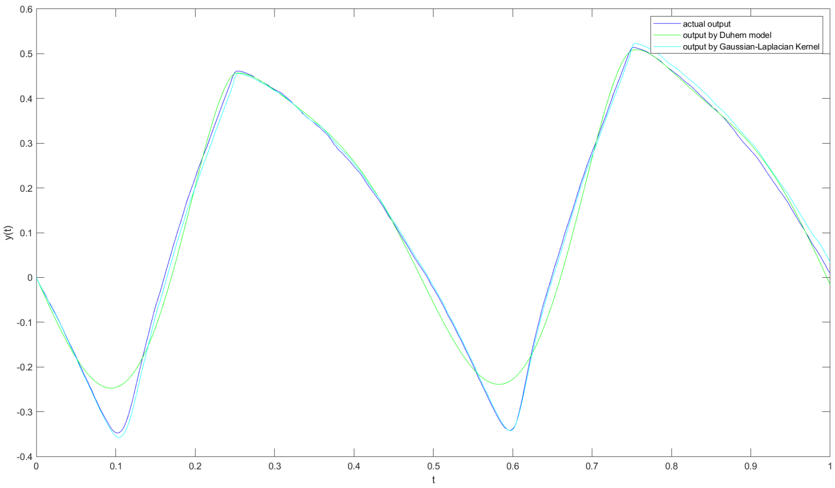


Figure 27: The output of the data, the output of the Duhem butterfly model the output of the combination of the Gaussian kernel and the Laplacian kernel method compared.

In Figure 28 the input-output behaviour of all studied methods together with the real input-output data can be found.

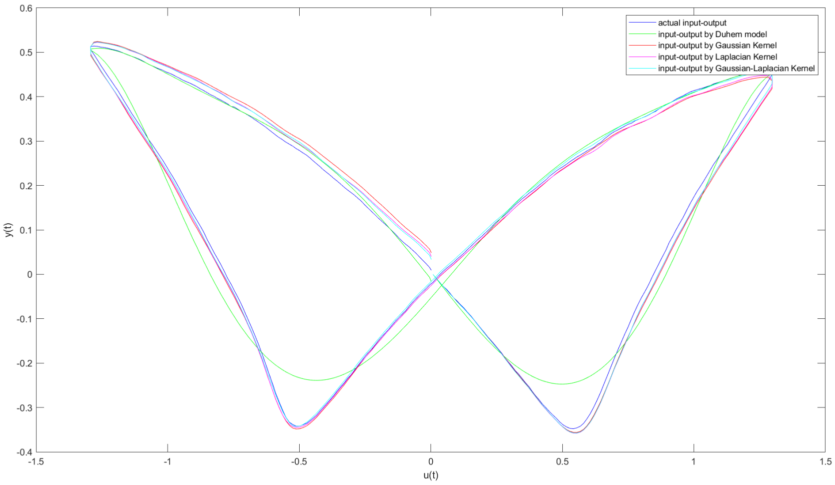


Figure 28: The input-output behaviour of the data, the Duhem butterfly model, The Gaussian kernel method, the Laplacian kernel method and the combination of the Laplacian and Gaussian kernel method compared.

By calculating the error, a value of 0.2495 is found. This is lower than the error of the Gaussian kernel function and the error of the Laplacian kernel function. If one looks closer to the lower wing of the butterfly, the difference in the methods can be studied more closely. This can be seen in Figure 29.

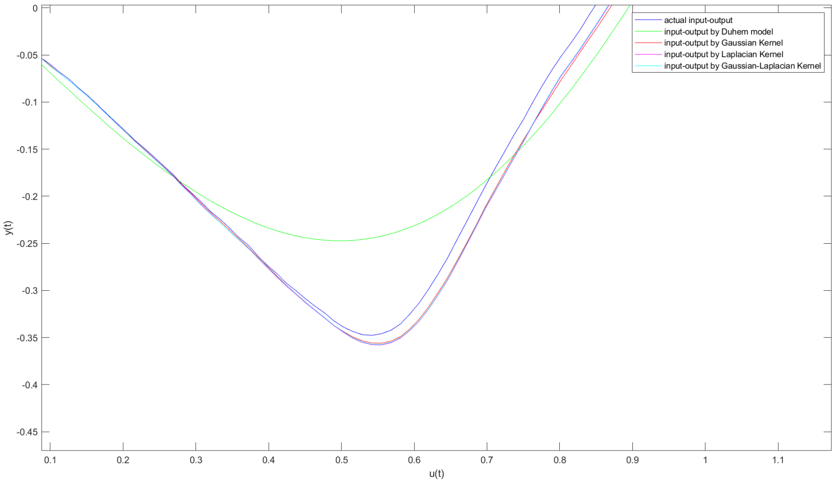


Figure 29: The input-output behaviour of the data, the Duhem butterfly model, the Gaussian kernel method, the Laplacian kernel method, and the combined Gaussian-Laplacian kernel method zoomed-in on the lower wing of the butterfly.

From this figure, it can be concluded that the kernel methods are better approximations of the obtained data than the fifth-order Duhem butterfly model. If the different kernel methods are compared, it can be seen that the linear combination of the Gaussian and Laplacian kernel functions gives the smallest error and therefore is the best method.

5 Conclusion

The hysteretic deformable mirror is a deformable mirror that exhibits hysteresis due to the piezoelectric-based actuators. To describe hysteretic phenomena, hysteresis models are defined. An operator is called a hysteresis operator if it is both causal and rate-independent. When a certain voltage is applied to the HDM, the input-output behaviour describes butterfly behaviour. An operator is called a butterfly hysteresis operator if there exists a hysteresis loop for which the signed area of the clockwise oriented loop equals the signed area of the counterclockwise oriented loop. Phenomenological models focus on the empirical description of input-output behaviour. Two commonly used phenomenological models are the Preisach butterfly operator and the Duhem butterfly operator.

To find mathematical models of dynamical systems from observed input-output data, identification is needed. The classical methods, such as parametric prediction error methods, focus on minimizing the difference between the output given by the data and the output given by the model. One can minimize this difference by finding an optimal parameter estimate. In the case of the HDM, the expected output is described by the Bouc-Wen model. The prediction error method has some advantages as it can be used in closed-loop systems and it has good asymptotic properties for many parametrizations. However, due to the for example high dimensionality, it also has some disadvantages such as overfitting and the bias-variance dilemma. From these two disadvantages, it can be an issue to choose the right model order. When one chooses incorrectly, this might lead to an ill-posed problem or a less robust model.

An identification method that keeps the well-posedness of the solution is needed. The identification method that satisfies the favorable properties is function estimation with the use of regularized kernel methods. In this case, the function g is searched in a reproducing kernel Hilbert space. The Hilbert space is completely determined by its reproducing kernel and, for modeling, a single positive semidefinite kernel function that encodes the desired properties of the function can be chosen. The difference between the prediction error method and the kernel method is that the number of parameters are not fixed a priori.

To make a comparison with the prediction error method, the impulse response estimation problem is considered. The solution to the objective functional in the impulse response estimation problem admits a finite-dimensional representation.

In the lab, an experiment with samples fabricated with Nb−doped PZT−ZrO_2, was executed. When a voltage with a maximum of 1300 Volt was applied, the samples showed indeed the desired butterfly behaviour. In order to identify the butterfly behaviour, both the prediction error method and the kernel method was used. For the prediction error method, the Duhem butterfly model with real data is considered. In this case, a polynomial up to order 5 was used with 6 different parameters. As the loop was nonsymmetric, two different series of parameters were chosen. One when the change over time of the input was negative and one when the change over time of the input was positive. Thereafter, the kernel method was applied. The Gaussian kernel function, the Laplacian kernel function, and a linear combination of the two of them were used to identify the data. After choosing the optimal parameters for the kernel functions, it turned out that the linear combination of the Gaussian kernel and the Laplacian kernel function was the best choice. This kernel function gave an error that was 3.5 times as small as the fifth-order Duhem butterfly model.

6 Discussion

This discussion starts with the validity and reliability of the research. Thereafter, suggestions for further research will be given.

The experiments have been performed with four different samples. To identify the system and make the identification more reliable, one would like to have more samples. In this way, the identification can be done with the identification step and thereafter the validation. This will give a more accurate description of the system. Furthermore, more data can help with choosing the right kernel.

In this research, prediction error methods up to order five are discussed. To make the model more accurate and make a comparison between the prediction error method and the classical method, one could increase the order to for example seven to make it more accurate. This comparison might be more valid as a more accurate model using the prediction error method is used.

There are some suggestions for future research. First of all, one could look into the kernels that are causal and rate-independent. If this is the case, those kernels could describe the hysteretic behaviour. Secondly, one could perform more experiments to identify and validate kernels that can be used to describe the mathematical behaviour of the HDM. Finally, one could look into identifying the differential equation that describes the hysteresis behaviour using kernel methods rather than identifying a part of this differential equation.

7 References

[1] Juan C Agüero, Graham C Goodwin, and Paul MJ Van den Hof. "A virtual closed loop method for closed loop identification". In: *Automatica* 47.8 (2011), pp. 1626–1637.

[2] Marco Augusto Vasquez Beltran. *Multi-loop Hysteresis and Recursive Remnant Control*. University of Groningen, 2022.

[3] Marco Vasquez Beltran, Bayu Jayawardhana, and Reynier Peletier. "Modeling and analysis of Duhem hysteresis operators with butterfly loops". In: *ArXiv* (2021).

[4] Michael Biehl. *The Shallow and the Deep: A biased introduction to neural networks and old school machine learning*. University of Groningen, 2022.

[5] E Chicurel-Uziel. "Single equation without inequalities to represent a composite curve". In: *Computer aided geometric design* 21.1 (2004), pp. 23–42.

[6] Alan H Fielding and John F Bell. "A review of methods for the assessment of prediction errors in conservation presence/absence models". In: *Environmental conservation* 24.1 (1997), pp. 38–49.

[7] Robert Huisman et al. "High pixel number deformable mirror concept utilizing piezoelectric hysteresis for stable shape configurations". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 7.2 (2021), pp. 029002–029002.

[8] Rodney A Kennedy and Parastoo Sadeghi. *Hilbert space methods in signal processing*. Cambridge University Press, 2013.

[9] Daniel T Larose. *Data mining and predictive analytics*. John Wiley & Sons, 2015.

[10] Lennart Ljung. "Prediction error estimation methods". In: *Circuits, Systems and Signal Processing* 21 (2002), pp. 11–21.

[11] Jack W Macki, Paolo Nistri, and Pietro Zecca. "Mathematical models for hysteresis". In: *SIAM review* 35.1 (1993), pp. 94–123.

[12] Eva Ostertagová. "Modelling using polynomial regression". In: *Procedia Engineering* 48 (2012), pp. 500–506.

[13] Ruiyue Ouyang, Vincent Andrieu, and Bayu Jayawardhana. "On the characterization of the Duhem hysteresis operator with clockwise input–output dynamics". In: *Systems & Control Letters* 62.3 (2013), pp. 286–293.

[14] Ruiyue Ouyang and Bayu Jayawardhana. "Stability analysis and controller design for a linear system with Duhem hysteresis nonlinearity". In: (2012), pp. 1676–1681.

[15] Vern I Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*. Vol. 152. Cambridge university press, 2016.

[16] Gianluigi Pillonetto et al. "Kernel methods in system identification, machine learning and function estimation: A survey". In: *Automatica* 50.3 (2014), pp. 657–682.

[17] Alex Simpkins. "System identification: Theory for the user, (ljung, l.; 1999)[on the shelf]". In: *IEEE Robotics & Automation Magazine* 19.2 (2012), pp. 95–96.

[18] Victor Sreeram and P Agathoklis. "On the properties of Gram matrix". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 41.3 (1994), pp. 234–237.

[19] Marco Augusto Vasquez-Beltran, Bayu Jayawardhana, and Reynier Peletier. "On the Characterization of Butterfly and Multiloop Hysteresis Behavior". In: *IEEE Transactions on Automatic Control* 67.7 (2021), pp. 3494–3506.

[20] Henk J van Waarde and Rodolphe Sepulchre. "Kernel-based models for system analysis". In: *IEEE Transactions on Automatic Control* (2022).

8 Appendix

Code belonging to Example 2.

```
clear all
close all

%Define time intervals
t1=1;
t2=2;
t3=3;
t4=4;
t5=5;

%Define input
umin= -1;
umax=-umin;

%Define time
t=[t1:0.01:t5];
size_t=size(t);
T=size_t(1,2);

%Define input
for i = 1:T
if t(i)>=t1 && t(i)<=t2
    u(i) = t(i)^2-2*t(i);
elseif t(i) >=t2 && t(i)<=t3
    u(i) = t(i)^2-4*t(i)+4;
elseif t(i) >=t3 && t(i)<=t4
    u(i) = -t(i)^2+6*t(i)-8;
elseif t(i) >=t4 && t(i)<=t5
    u(i) = -t(i)+4;
end
end

%Define output
for i=1:T
```

```
if t(i)>=t1 && t(i)<=t2
    P(i) = -(umax+u(i)).^2;
elseif t(i) >=t2 && t(i)<=t3
    P(i) = -(umax-u(i))*(umax+3*u(i));
elseif t(i) >=t3 && t(i)<=t4
    P(i) = -(umax-u(i)).^2;
elseif t(i) >=t4 && t(i)<=t5
    P(i) = -(umax+u(i))*(umax-3*u(i));
end
end

%Make a butterfly
figure;
plot(u,P)
title('Input-output phase plot of input and output
   signal')
xlabel('u(t)')
ylabel('P(t)')

%Input signal
figure;
plot(t,u)
title('Input signal u in periodic time interval [t_1
   , t_5]')
xlabel('t')
ylabel('u(t)')
```

Code belonging to Example 3.

```
%This ODE belongs to BoucWenmodel.m
function dydt = ODE_system(t, y, n)
    u_prime_piecewise = @(t) cos(t);
    dydt = u_prime_piecewise(t) - 2 * abs(y)^n *
       u_prime_piecewise(t) - y * abs(y)^(n - 1)*abs
       (u_prime_piecewise(t));
end

clear all
close all

%Define n as a natural number
n = 2;
t=0:0.01:20;
y0 = 0.5;

%Solve ODE in ODE_system
[t, y] = ode45(@(t, y) ODE_system(t, y, n), t, y0);

%Define input
umin= -1;
umax=-umin;

%Define time
t=0:0.01:20;
size_t=size(t);
T=size_t(1,2);

%Define input
for i = 1:T
u(i) = sin(t(i));
end

%Make plot
figure;
plot(u, y);
```

```
xlabel('u');
ylabel('y');
title('Plot of a continuous periodic input function
   and its output');
```

Code belonging to Example 4.

```
%This ODE belongs to duhembutterfly.m
function dydt = ODE_systemduhembutterfly(t, y)
    u = @(t) 5*sin(t);
    u_prime_piecewise = @(t) 5*cos(t);
    dydt = u_prime_piecewise(t) + 2*u(t)*
       u_prime_piecewise(t) + u(t)^3*
       u_prime_piecewise(t)- y*abs(u_prime_piecewise
       (t));
end

clear all
close all

%Define time for ODE
t=0:0.01:10;

%Define initial condition
y0 = -5;

%Solve ODE in ODE_systemduhembutterfly
[t, y] = ode45(@(t, y) ODE_systemduhembutterfly(t, y
   ), t, y0);

%Define input
umin= -5;
umax=-umin;

%Define time
t=0:0.01:10;
size_t=size(t);
T=size_t(1,2);

%Define input
for i = 1:T
u(i) = 5*sin(t(i));
end
```

```
% Make a butterfly
figure;
plot(u, y);
xlabel('u');
ylabel('y');
title('Plot of a piecewise continuous input function
    and its output');
```

Code belonging to Section 4.1

```
T = readtable('C:\Users\lisah\Downloads\Data
   Experiment 2 1.3kV.txt');

%Import data
time = T{:,1};
input = T{:,2};
output1 = T{:,10};

%Plots
plot(time,input)
xlim([0 1]);
xticks(0:0.25:1);
ylim([-1300 1300]);
xlabel('t');
ylabel('u(t)');
title('Applied input function');

plot(time, output1)
xlim([0 1]);
xticks(0:0.25:1);
xlabel('t');
ylabel('y(t)');
title('Obtained output function');

plot(input, output1)
xlim([-1300 1300]);
xlabel('u(t)');
ylabel('y(t)');
title('The input output behaviour');
```

Code belonging to Section 4.2.1

```
close all
close all
T = readtable('C:\Users\lisah\Downloads\Data
   Experiment 2 1.3kV.txt');

%Import data
time = T{:,1};
input = T{:,2}./1000;
output1 = T{:,10}./1000;
%output2=T{:,11};
%output3=T{:12};

% subplot(2,2,1)
% plot(time,input)
% subplot(2,2,2)
% plot(input, output1)


%Data fit
dudt = gradient(input(:))./gradient(time(:));
dydt = gradient(output1(:))./gradient(time(:));

%Define parameters
syms a0 a1 a2 a3 a4 a5 a6

%Positive/Negative \dot{u}
index = find(dudt<0);

%Solve least squares
Alpha=[(dudt(index))';
    (input(index).*dudt(index))';
    (input(index).^2.*dudt(index))';
    (input(index).^3.*dudt(index))';
    (input(index).^4.*dudt(index))' ;
    (input(index).^5.*dudt(index))']';
beta=dydt(index)-output1(index).*dudt(index);
```

```
%xi= [a0;a1;a2;a3;a4;a5];
xi= pinv(Alpha)*beta;


%Start solving ODEs
%First positive \dot{u}
t = [0:0.0025:0.25];
y0 = 0;
[t, y1] = ode45(@(t, y) ODE_data_xi_5_pos1(t, y), t,
    y0);

%Negative \dot{u}
t = [0.25:0.0025:0.75];
y0 = y1(101);
[t,y2] = ode45(@(t, y) ODE_data_xi_5_neg(t, y), t,
   y0);

%Second positive \dot{u}
t = [0:0.0025:0.25];
y0 = y2(201);
[t, y3] = ode45(@(t, y) ODE_data_xi_5_pos1(t, y), t,
    y0);

y = [0;y1(1:100);y2(1:200);y3(1:100)];


%plot(time,output1,':')
%hold on
%plot(t, y,'--')
%legend('actual output','output by Duhem model')
%xlabel('t');
%ylabel('y(t)');

plot(input,output1,':')
hold on
plot(input, y, '--')
xlabel('u(t)');
```

```
ylabel('y(t)');
legend('actual input-output','input-output by Duhem
   model')

norm(output1-y);
```

Code belonging to Section 4.2.2

Gaussian Kernel

```
close all
close all
T = readtable('C:\Users\lisah\Downloads\Data
   Experiment 2 1.3kV.txt');

%Import data
time = T{:,1};
input = T{:,2}/1000;
output1 = T{:,10}/1000;

dudt = gradient(input(:))./gradient(time(:));
dydt = gradient(output1(:))./gradient(time(:));

%index = find(dudt <0)
inputneg = input(101:300);
inputpos = [input(1:100);input(301:400)];

outputneg = output1(101:300);
outputpos = [output1(1:100);output1(301:400)];

dudtneg = dudt(101:300);
dudtpos = [dudt(1:100);dudt(301:400)];

dydtneg = dydt(101:300);
dydtpos = [dydt(1:100);dydt(301:400)];

%plot(time,dudt);

hatfneg = dydtneg./dudtneg - outputneg;
hatfpos = dydtpos./dudtpos - outputpos;

%Gaussian Kernel
sigma = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2));
```

```
%Case: \dot{u} is negative
Gneg=zeros(200);
for i = 1: 200
    for j = 1:200
Gneg(i,j) = k(inputneg(i),inputneg(j));
    end
end

gamma = 0.00000001;


%dit zijn de c_j's
Cneg = inv(Gneg+gamma*eye(200))*hatfneg;

x_neg = inputneg;
y_neg = zeros(200,1);

for j = 1:200
sum = 0;
for i = 1:200
    sum = sum + k(x_neg(j),inputneg(i))*Cneg(i);
end
y_neg(j) = sum;
end

%Case: \dot{u} is positive
Gpos=zeros(200);
for i = 1: 200
    for j = 1:200
Gpos(i,j) = k(inputpos(i),inputpos(j));
    end
end

gamma = 0.00000001;

%dit zijn de c_j's
Cpos = inv(Gpos+gamma*eye(200))*hatfpos;
```

```
x_pos = inputpos;
y_pos = zeros(200,1);

for j = 1:200
sum = 0;
for i = 1:200
    sum = sum + k(x_pos(j),inputpos(i))*Cpos(i);
end
y_pos(j) = sum;
end

t1 = [0:0.0025:0.25];
y0 = 0;
[t1, ykernel1] = ode45(@(t, y)
   ODE_data_kernel_Gaussian_pos1(t, y,Cpos, inputpos
   ), t1, y0);

t2 = [0.25:0.0025:0.75];
y1 = ykernel1(101);
[t2, ykernel2] = ode45(@(t, y)
   ODE_data_kernel_Gaussian_neg(t, y,Cneg, inputneg)
   , t2, y1);

t3 = [0.75:0.0025:1];
y2 = ykernel2(201);
[t3, ykernel3] = ode45(@(t, y)
   ODE_data_kernel_Gaussian_pos2(t, y,Cpos, inputpos
   ), t3, y2);

ykernel = [ykernel1(1:101);ykernel2(2:201);ykernel3
   (2:101)];
t = [t1;t2(2:201);t3(2:101)];

%plot(t,ykernel)
%plot(input,ykernel)

error = norm(ykernel-output1)
```

```
plot(input,output1,"b")
hold on
plot(input, y,"g")
hold on
plot(input, ykernel,"r")
xlabel('u(t)');
ylabel('y(t)');
legend('actual input-output','input-output by Duhem
   model', 'input-output by Gaussian Kernel')

close all
function dydt5 = ODE_data_kernel_Gaussian_pos1(t,y,
   Cpos, inputpos)
u = @(t) 5.2*t;
u_dot = 5.2;

sigma = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2));

    function sum = fhat_pos(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputpos(i))*Cpos(i);
    end
    end

dydt5 = fhat_pos(u(t))*u_dot + y*u_dot;

end

close all
function dydt5 = ODE_data_kernel_Gaussian_pos2(t,y,
   Cpos, inputpos)
u = @(t) 5.2*t-5.2;
u_dot = 5.2;

sigma = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2));
```

```
    function sum = fhat_pos(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputpos(i))*Cpos(i);
    end
    end

dydt5 = fhat_pos(u(t))*u_dot + y*u_dot;

end

close all
function dydt5 = ODE_data_kernel_Gaussian_neg(t,y,
   Cneg, inputneg)
u = @(t) -5.2*t+2.6;
u_dot = -5.2;

sigma = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2));

    function sum = fhat_neg(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputneg(i))*Cneg(i);
    end
    end

dydt5 = fhat_neg(u(t))*u_dot + y*u_dot;

end
```

Laplacian Kernel

```
close all
T = readtable('C:\Users\lisah\Downloads\Data
   Experiment 2 1.3kV.txt');

%Import data
time = T{:,1};
input = T{:,2}/1000;
output1 = T{:,10}/1000;

dudt = gradient(input(:))./gradient(time(:));
dydt = gradient(output1(:))./gradient(time(:));

%index = find(dudt <0)
inputneg = input(101:300);
inputpos = [input(1:100);input(301:400)];

outputneg = output1(101:300);
outputpos = [output1(1:100);output1(301:400)];

dudtneg = dudt(101:300);
dudtpos = [dudt(1:100);dudt(301:400)];

dydtneg = dydt(101:300);
dydtpos = [dydt(1:100);dydt(301:400)];

%plot(time,dudt);

hatfneg = dydtneg./dudtneg - outputneg;
hatfpos = dydtpos./dudtpos - outputpos;

%Laplacian Kernel
rho = 0.0005;
k = @(x,y) rho*(exp(-norm(x-y)));

%Case: \dot{u} is negative
Gneg=zeros(200);
```

```
for i = 1: 200
    for j = 1:200
Gneg(i,j) = k(inputneg(i),inputneg(j));
    end
end

gamma = 0.00000001;


%dit zijn de c_j's
Cneg = inv(Gneg+gamma*eye(200))*hatfneg;

x_neg = inputneg;
y_neg = zeros(200,1);

for j = 1:200
sum = 0;
for i = 1:200
    sum = sum + k(x_neg(j),inputneg(i))*Cneg(i);
end
y_neg(j) = sum;
end

%Case: \dot{u} is positive
Gpos=zeros(200);
for i = 1: 200
    for j = 1:200
Gpos(i,j) = k(inputpos(i),inputpos(j));
    end
end

gamma = 0.00000001;

%dit zijn de c_j's
Cpos = inv(Gpos+gamma*eye(200))*hatfpos;

x_pos = inputpos;
y_pos = zeros(200,1);
```

```
for j = 1:200
sum = 0;
for i = 1:200
    sum = sum + k(x_pos(j),inputpos(i))*Cpos(i);
end
y_pos(j) = sum;
end

t1 = [0:0.0025:0.25];
y0 = 0;
[t1, ykernel4] = ode45(@(t, y)
   ODE_data_kernel_Laplacian_pos1(t, y,Cpos,
   inputpos), t1, y0);

t2 = [0.25:0.0025:0.75];
y1 = ykernel4(101);
[t2, ykernel5] = ode45(@(t, y)
   ODE_data_kernel_Laplacian_neg(t, y,Cneg, inputneg
   ), t2, y1);

t3 = [0.75:0.0025:1];
y2 = ykernel5(201);
[t3, ykernel6] = ode45(@(t, y)
   ODE_data_kernel_Laplacian_pos2(t, y,Cpos,
   inputpos), t3, y2);

ykernellaplace = [ykernel4(1:101);ykernel5(2:201);
   ykernel6(2:101)];
t = [t1;t2(2:201);t3(2:101)];

% plot(t, ykernellaplace,"r")
% xlabel('t');
% ylabel('y(t)');

% plot(t,output1,"b")
% hold on
% plot(t, y,"g")
% hold on
% plot(t,ykernel, "r")
% hold on
% plot(t, ykernellaplace,"m")
% xlabel('t');
% ylabel('y(t)');
% legend('actual output','output by Duhem model', '
   output by Gaussian Kernel', 'output by Laplacian
   Kernel')


plot(input,output1,"b")
hold on
plot(input, y,"g")
hold on
plot(input,ykernel, "r")
hold on
plot(input, ykernellaplace,"m")
xlabel('u(t)');
ylabel('y(t)');
legend('actual input-output','input-output by Duhem
   model', 'input-output by Gaussian Kernel', 'input
   -output by Laplacian Kernel')

function dydt5 = ODE_data_kernel_Laplacian_pos1(t,y,
   Cpos, inputpos)
u = @(t) 5.2*t;
u_dot = 5.2;

k = @(x,y) 0.0005*(exp(-norm(x-y)));

    function sum = fhat_pos(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputpos(i))*Cpos(i);
    end
    end
```

```
dydt5 = fhat_pos(u(t))*u_dot + y*u_dot;

end

function dydt5 = ODE_data_kernel_Laplacian_pos2(t,y,
   Cpos, inputpos)
u = @(t) 5.2*t-5.2;
u_dot = 5.2;

k = @(x,y) 0.0005*(exp(-norm(x-y)));

    function sum = fhat_pos(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputpos(i))*Cpos(i);
    end
    end

dydt5 = fhat_pos(u(t))*u_dot + y*u_dot;

end

function dydt5 = ODE_data_kernel_Laplacian_neg(t,y,
   Cneg, inputneg)
u = @(t) -5.2*t+2.6;
u_dot = -5.2;

k = @(x,y) 0.0005*(exp(-norm(x-y)));

    function sum = fhat_neg(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputneg(i))*Cneg(i);
    end
    end

dydt5 = fhat_neg(u(t))*u_dot + y*u_dot;
```

```
end
```

Laplacian and Gaussian Kernel combined

```
close all
T = readtable('C:\Users\lisah\Downloads\Data
   Experiment 2 1.3kV.txt');

%Import data
time = T{:,1};
input = T{:,2}/1000;
output1 = T{:,10}/1000;

dudt = gradient(input(:))./gradient(time(:));
dydt = gradient(output1(:))./gradient(time(:));

%index = find(dudt <0)
inputneg = input(101:300);
inputpos = [input(1:100);input(301:400)];

outputneg = output1(101:300);
outputpos = [output1(1:100);output1(301:400)];

dudtneg = dudt(101:300);
dudtpos = [dudt(1:100);dudt(301:400)];

dydtneg = dydt(101:300);
dydtpos = [dydt(1:100);dydt(301:400)];

%plot(time,dudt);

hatfneg = dydtneg./dudtneg - outputneg;
hatfpos = dydtpos./dudtpos - outputpos;

%Gaussian + Laplacian Kernel
sigma=0.11;
rho = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2)) + rho*(exp(-
   norm(x-y)));
```

```
%Case: \dot{u} is negative
Gneg=zeros(200);
for i = 1: 200
    for j = 1:200
Gneg(i,j) = k(inputneg(i),inputneg(j));
    end
end

gamma = 0.00000001;


%dit zijn de c_j's
Cneg = inv(Gneg+gamma*eye(200))*hatfneg;

x_neg = inputneg;
y_neg = zeros(200,1);

for j = 1:200
sum = 0;
for i = 1:200
    sum = sum + k(x_neg(j),inputneg(i))*Cneg(i);
end
y_neg(j) = sum;
end

%Case: \dot{u} is positive
Gpos=zeros(200);
for i = 1: 200
    for j = 1:200
Gpos(i,j) = k(inputpos(i),inputpos(j));
    end
end

gamma = 0.00000001;

%dit zijn de c_j's
Cpos = inv(Gpos+gamma*eye(200))*hatfpos;
```

```
x_pos = inputpos;
y_pos = zeros(200,1);

for j = 1:200
sum = 0;
for i = 1:200
    sum = sum + k(x_pos(j),inputpos(i))*Cpos(i);
end
y_pos(j) = sum;
end

t1 = [0:0.0025:0.25];
y0 = 0;
[t1, ykernel7] = ode45(@(t, y)
   ODE_data_kernel_GausLaplac_pos1(t, y,Cpos,
   inputpos), t1, y0);

t2 = [0.25:0.0025:0.75];
y1 = ykernel7(101);
[t2, ykernel8] = ode45(@(t, y)
   ODE_data_kernel_GausLaplac_neg(t, y,Cneg,
   inputneg), t2, y1);

t3 = [0.75:0.0025:1];
y2 = ykernel8(201);
[t3, ykernel9] = ode45(@(t, y)
   ODE_data_kernel_GausLaplac_pos2(t, y,Cpos,
   inputpos), t3, y2);

ykernelGausLaplac = [ykernel7(1:101);ykernel8(2:201)
   ;ykernel9(2:101)];
t = [t1;t2(2:201);t3(2:101)];



% plot(t, ykernelGausLaplac,"r")
% xlabel('t');
% ylabel('y(t)');
% plot(t,output1,"b")
% hold on
% plot(t, y,"g")
% hold on
% plot(t,ykernel, "r")
% hold on
% plot(t, ykernellaplace,"m")
% plot(t,ykernelGausLaplac,"c")
% xlabel('t');
% ylabel('y(t)');
% legend('actual output','output by Duhem model', '
   output by Gaussian Kernel', 'output by Laplacian
   Kernel', 'output by Gaussian-Laplacian Kernel' )



plot(input,output1,"b")
hold on
plot(input, y,"g")
hold on
plot(input,ykernel, "r")
hold on
plot(input, ykernellaplace,"m")
hold on
plot(input,ykernelGausLaplac,"c")
xlabel('u(t)');
ylabel('y(t)');
legend('actual input-output','input-output by Duhem
   model', 'input-output by Gaussian Kernel', 'input
   -output by Laplacian Kernel', 'input-output by
   Gaussian-Laplacian Kernel')


error = norm(ykernelGausLaplac-output1);

function dydt5 = ODE_data_kernel_GausLaplac_pos1(t,y
   ,Cpos, inputpos)
```

```
u = @(t) 5.2*t;
u_dot = 5.2;

sigma=0.11;
rho = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2)) + rho*(exp(-
   norm(x-y)));

    function sum = fhat_pos(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputpos(i))*Cpos(i);
    end
    end

dydt5 = fhat_pos(u(t))*u_dot + y*u_dot;

end

function dydt5 = ODE_data_kernel_GausLaplac_pos2(t,y
   ,Cpos, inputpos)
u = @(t) 5.2*t-5.2;
u_dot = 5.2;

sigma=0.11;
rho = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2)) + rho*(exp(-
   norm(x-y)));

    function sum = fhat_pos(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputpos(i))*Cpos(i);
    end
    end

dydt5 = fhat_pos(u(t))*u_dot + y*u_dot;
```

```
end

function dydt5 = ODE_data_kernel_GausLaplac_neg(t,y,
   Cneg, inputneg)
u = @(t) -5.2*t+2.6;
u_dot = -5.2;

sigma=0.11;
rho = 0.01;
k = @(x,y) (exp(-norm(x-y)^2/sigma^2)) + rho*(exp(-
   norm(x-y)));

    function sum = fhat_neg(x)
        sum = 0;
    for i = 1:200
    sum = sum + k(x,inputneg(i))*Cneg(i);
    end
    end

dydt5 = fhat_neg(u(t))*u_dot + y*u_dot;

end
```