UNIVERSITY OF GRONIGEN

FACULTY OF SCIENCE AND ENGINEERING

ASTRONOMY: INTRUMENTATION AND INFORMATICS

Controlling the Remnant Deformation of a Piezoelectric Actuator Using A Root Finding Algorithm

Master's Thesis
P. B. (Pedro) MENDES

Supervisor:
Prof. Dr. B (Bayu)
JAYAWARDHANA

September 2019

rijksuniversiteit groningen

1,500 years ago, everybody knew that the Earth was the center of the universe. 500 years ago, everybody knew that the Earth was flat. And 15 minutes ago, you knew that humans were alone on this planet. Imagine what you'll know tomorrow."

From the movie "Men in Black"

O homem primeiro tropeça, depois anda, depois corre, um dia voará.

José Saramago," Memorial do Convento"

[Blank Page]

Abstract

As humans, we became a curious species, and one of the biggest targets of that curiosity is the distant Universe around us. One of the questions we started asking ourselves is: are there any other planets like Earth? With the amount of stars in the Milky Way, other star systems like our own Solar System might exist. Recently, technology started to evolve in such a way we can start observing these stellar systems with very powerful telescopes. However, this is still a somewhat "recent" technology, and one of the challenges is to develop reliable deformable mirrors capable of correcting the wavefront that arrives to us from these distance systems. The Netherlands Institute for Space Research recently proposed a new design for a deformable mirror that makes use of the hysteretic nature and properties of a chosen material to increase the performance of these deformable mirrors. This new approach relies on the control of the remnant deformation of such material, to which the corresponding deformation loop might not even be a single hysteretic loop, but a butterfly loop. These loops are modeled using non-linear models, which, together with the shape of the butterfly loop itself, makes the controlling a challenging step to the new design. This thesis will present a new approach for a controlling algorithm for the remnant deformation of these actuators.

[Blank Page]

Acknowledges

I would like to first thanks the HDM group for receiving me and giving me the opportunity to work in such an interesting and potentially important project. Professor Dr. Bayu Jayawardhana and Dr. Robert Huisman in particular, for being the ones who have been looking over me and advising me during not only my internship but also my master project. I would also like to thank Marco Vasquez for helping me brainstorming when starting this project, and for the help you gave me along the way.

Se esta tese representa o culminar de anos de estudos, é impossível eu não mencionar os professores que ao longo de toda a minha vida como estudante me marcaram e influenciaram de uma forma ou outra a chegar até aqui. Professores como as professoras Ermelinda, Rosário, Ana Mourão, Ivone Amaro, Lucília Brito e Constança Providência, e os professores José Alves, João Sá, Carlos Boavida, e Fernando Nogueira. Foram modelos do poder do saber e de onde até a sede de conhecimento pode levar. Não chegarei a ser Presidente, como um de vocês me disse há largos anos, mas espero chegar a ser um alguém que faça justiça ao bom exemplo que me passaram.

To my fellow Golfbrekers, thank you so much for receiving me with open arms on your pool. These 2 last seasons were incredible, and it is hard to explain how much being able to swim with you and for you helped me with my self-esteem during all this time. From the 100% real dutchies, to the exotic internationals, from lane 1 to lane 6, you are the best dam student swimming team one could wish for. Go blue and yellow, go Golfbrekers! (Do I need to say more?)

Muito obrigado aos Tugas em Groningen, porque quando se está longe de casa, é sempre bom poder sentir um pouco do que é familiar. Destes, há que mencionar o Pai, Zé Rui. Muito obrigado, não só por seres a figura paternal deste grupo, mas por também me teres dado a oportunidade de estagiar contigo, e de poder ter o meu primeiro contacto com o que é trabalhar para o Espaço. Se uma parte de mim for para fora deste planeta, ou ate ao limite de tal, será graças à oportunidade que me deste. E boa sorte com o balão!

A toda a malta do DF a quem pude chamar de amigo. Ser estudante em Coimbra só tem o seu encanto se tivermos colegas com quem partilhar esse encanto. À malta do carro 2012/2013, porque 3 anos ao vosso lado levam histórias para a vida. A toda a família Cosa Nostra, com um abraço especial ao Padrinho Trenk, ao avô Ricardo, e aos 7 afilhados, uma linhagem exemplar.

Às melhores amigas de diferentes fases da minha vida: Adriana, Beatriz, Joana, Luísa e Nicole. Entre conversas de autocarros, intervalos de aulas, pausas nas Químicas, partilhas de piadas horríveis e desabafos de trios ou duos, estiveram ao meu lado e compreenderam-me como pouca gente me sequer tentou compreender. Longe de lugares e até de tempo, serão para sempre as melhores.

À minha segunda família, a secção de natação da AAC. Se acredito que só sou quem sou devido à natação, todos os que acompanharam a minha "carreira" como nadador fizeram parte desse processo de crescimento. Já vão 12 anos desde que me juntei à Briosa, e espero poder continuar, de alguma forma, a poder representar o emblema ao peito. Às gerações de 94/95, que durante tantos anos contámos azulejos juntos, e com alguns de nós ainda a dar braçadas, contrário ao que seria de esperar. Às diferentes gerações de direções que nos deram as condições para podermos mostrar o nosso valor, com uma especial menção ao nosso Senhor Jorge. Aos passados treinadores que me moldaram a ser um melhor atleta e um melhor homem: Anabela e Guerra, estes não na AAC, mas que ainda assim me puseram o "bichinho" da natação e da competição; Abrantes e Migas, que me formaram e me integraram na equipa da melhor forma possível. E àquele que considero como um segundo pai: o Mister Fausto Pinto Ângelo. Seria necessário uma outra dissertação para te dizer o que fizeste por mim e o quão importante és. Mas espero que ao fim de todos este anos, já saibas isso. Se não, simplesmente continuarei a oferecer alheiras no Natal, como forma de agradecimento.

Em último lugar, gostava de puder agradecer a toda a minha família. Estar longe de casa não é fácil, mas seria pior se não soubesse que cada vez que volto sou recebido de braços abertos por aqueles que me amam e que perguntam sempre como ando, naquelas terras Holandesas. Aos meus avós, agradeço por me terem mostrado o que é ser humilde, e que sucesso na vida, seja esse em qualquer forma que queiramos, não depende de onde começamos, mas sim do que fazemos com o que nos é dado. Gostava que ambas as minhas avós pudessem presenciar este meu feito, porque sem elas a criança que fazia bolos de terra e cantava lengalengas não estaria aqui.

Aos meus pais e irmão, eu não agradeço, eu dedico esta tese. Vocês são uma parte de mim, e portanto esta tese é vossa também. Sem a vossa ajuda e sacrifício, sem a vossa fé em mim, nada do que realizo agora seria remotamente possível. São a força que me move para chegar mais longe, e apenas espero ter vos feito orgulhosos.

[Blank Page]

[Blank Page]

Contents

[Blank Page]

[Blank Page]

1 - Introduction

A great effort has been made in the past decades to detect and study planets in stellar systems of the Milky Way. These are called Exoplanets, and recently many projects and instruments have been developed to survey these objects. Almost 4000 of these planets have been confirmed to exist, with many more waiting for confirmation, from projects like the Kepler and TESS missions being some of the most recent ones [1].

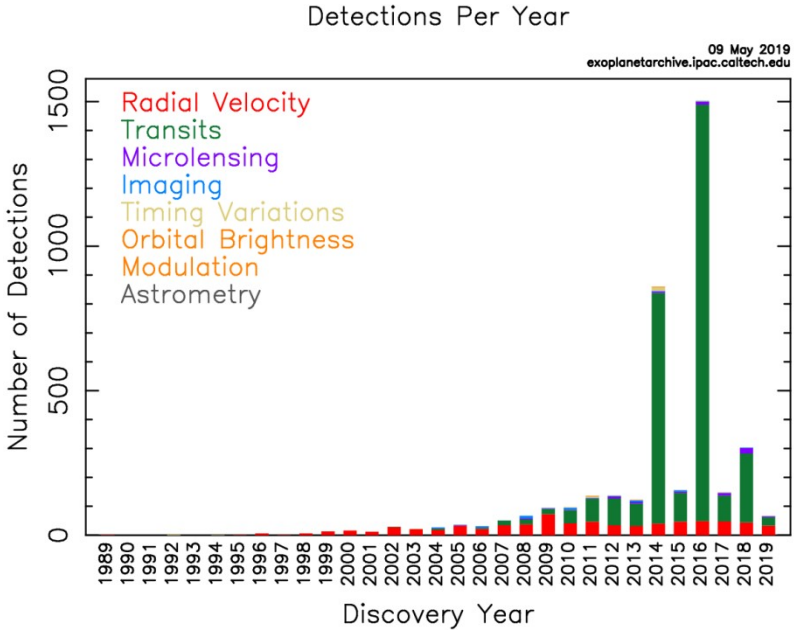


Figure 1 - Number of exoplanets discovered by year, and by detection technique.

However, very few of the exoplanets detected have been observed by direct imaging. To the present day, the two most used methods for exoplanet detection have been radial velocity and transits [2], as seen in Figure 1. Radial velocity techniques rely on the effect that the orbiting planet has on the star position, which follows a small orbit due to the gravity of the planet. The motion of the star is very small, but it can be detected from Earth, and by calculating how that star position changes relative to Earth, we can obtain properties like the size and orbit radius of the planet(s) that orbits the start. On transit techniques, the luminosity of the start is observed, and periodic changes in luminosity indicates the moments when a planet passes in front of the star and blocks part of its light. This method can also give us important information about the size and orbit of the planet(s). Figure 2 shows the application of these techniques.

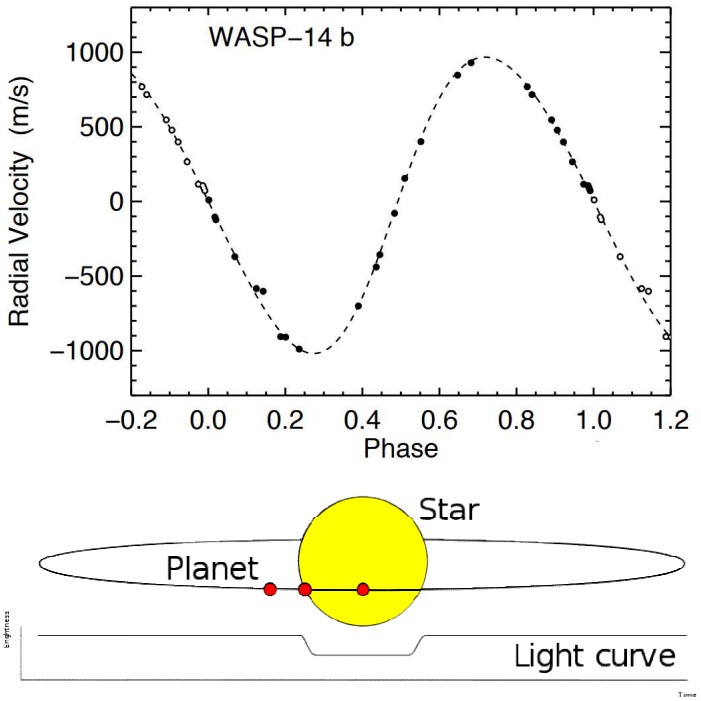


Figure 2 - On top, effect on the WASP-14 start system on the radial velocity by its exoplanet. On bottom, example on how a exoplanet can affect the perceiving luminosity of the hosting star.

Although indirect methods have played a crucial role in the detection of exoplanets and their stellar system, the properties obtained from these are often limited to size and mass of the planet(s) and radius of it(s) orbits. The desire to better understand exoplanets means that other methods have to be considered, and the best scenario would be to be able to do direct imaging of these planets. Only with direct imaging one can study other properties of the planet, such has the composition of its atmosphere, soil and core. Unfortunately, direct imaging methods for exoplanet detection depend on the recent imaging technologies, and most of them are still being developed and improved. Only in cases where the planet is big enough to be observed, close to the size of Jupiter, reliable direct imaging has been achieved. 2M1207b, the first exoplanet observed by direct imaging, is a example of such large planets [3].

There are 2 main problems that must be overcome to be able to use direct imaging for exoplanet detection [4]:

- Firstly, the exoplanets and their stellar systems are extremely far from Earth, and so the angular separation between the hosting star and its planets can be as small as 0.1"
- Secondly, since planets do not have proper light, the contrast ration between the brightness of the hosting start and its planet can be as low as 10^{-10}, meaning that the planets will be obfuscated by the star's light.

Finding a solution to these problems has proven to be a very interesting and big engineering challenge over the last years, and even in existing instruments, these problems have not been solved to a degree of satisfaction. Future projects demand for a better performance of optical instruments, and this performance can be achieved by the use of Adaptive Optic Systems (AOS).

1.1 - Coronagraph and Deformable Mirror (DM)

The coronagraph was first introduced in 1939 by Bernard Lyot [5]. Its initial purpose was to be able to study the corona ("atmosphere") of the sun, by blocking the light that directly came from it, but the same principle can be used to observe exoplanets near a start. In Figure 3, it is shown a simplified scheme of a coronagraph.

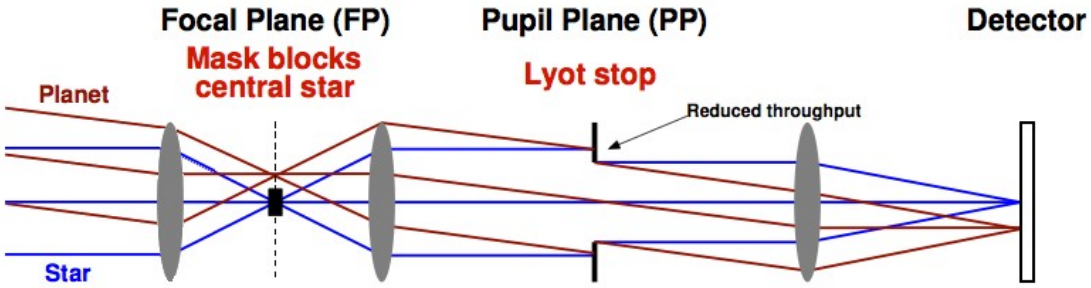


Figure 3 - Simplified schematic of a coronagraph.

After the first lens, a mask is placed in the focal point of that same lens. If the coronagraph instrument is aligned with the star (blue in image), the light from the star will be blocked by such mask. Any ray of light with a small angular deviation from that direction, for example, the light coming from the orbiting planet (red in image), will not hit the focal point, and so it will pass through.

The coronagraph was developed taking into account the sun/corona system, where the angular separation is not a big problem. But for exoplanet detection, this is a crucial point, and any ray of light from the hosting start that slightly deviates from the observing direction will block the planet's light. Due to the distance of the stellar system, scattering of light before it reaches the instrument is highly probable, and the instrument itself can scatter light as well. Furthermore, atmospheric scattering happens for ground-based telescopes. All these scattering processes will prevent a plane wavefront to reach the sensors, and so some of the star's light will pass through the mask, and block the planet's light. Wavefront correction must then be implemented with the use of a Adaptive Optics Systems (AOS) based on a deformable mirror (DM).

In Figure 4, a simple visual explanation of the function of the deformable mirror is shown. Because of the distance of the star system, the light that reach the instrument can be seen as a wavefront. Due to the previous mentioned refraction sources, the wavefront will not be plain, and so it will scatter when reaching the lens. If we apply a DM that copies the deformation of the wavefront, we will have a corrected plane wavefront, and so all the light from the star will be blocked by the coronagraph's mask.

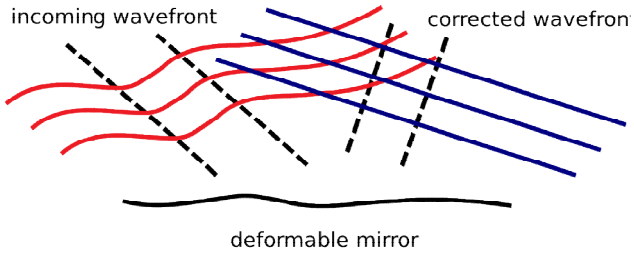


Figure 4 - Wavefront correction using a deformable mirror.

There are several designs for a DM used for wavefront correction, being the piston-based DM's the more simple and versatile design. However, for high definition imaging, the high number of actuators increases the complexity of the DM substantially, as well as the spacing between them. Current missions for exoplanet detection, such as WFIRST, have shown to use DM's with arrays of 64×64 actuators (4096 in total) [6], with DM's dimensions that can be just around some centimeters. This has shown to be a big challenge for the design of suitable DM's, not only due to the complexity of the bulky hardware needed to control each actuator individually, but also due to the large amount of energy necessary to operate the complete DM.

1.2 - Presenting a solution for deformable mirror: HDM

To overcome the challenges that current designs of deformable mirror present, the Netherlands Institute for Space Research has proposed new concept for such instrument: an hysteresis deformable mirror (HDM) [7]. Such mirror will make use of the properties of a piezoelectric material to solve some of these problems.

1.2.1 - Piezoelectricity

Piezoelectricity was first described by the French brothers Pierre and Jacques Curie, in 1880 [8]. The piezoelectric materials have the interesting property of, when applied mechanical stress, an electric field is generated [9]. The inverse process is also true: when an electrical field is applied in a certain direction of the material, it will deform. By having a model for the deformation of the give piezoelectric material, it would be possible to control the height of a thin layer of the material by inputting the correct potential in such direction. This is the reason why piezoelectric material have a very promising application for deformable mirror. However, there is an important phenomenon occurring in these materials to which the DM technology can further improve its performance: hysteresis.

1.2.2 - Hysteresis

Hysteresis was first described by Sir James Alfred Ewing [10]. In his works, Ewing observed that the relation between the magnetization of a ferroelectric material and the magnetic field applied between the limits of the material was not linear, and even had a "lagging" effect, meaning that when applying the magnetic field to the material and removing it, it will result in a different magnetization from the one it originally had when no magnetic field was present: the material has memory. In Figure 5 one of the first registered figures of an hysteretic behavior of a material made by Ewing.

Piezoelectric materials can also have this behavior: the relation between the applied electric field and the deformation of the material might also be not linear and will depend on past states of the material. The resulted hysteresis will allow us to use two more concepts that will further improve the design of the HDM: remnant deformation and multiplexing.

A schematic example of the remnant deformation is illustrated in Figure 6. To hold the material at a certain thickness, there is no need to apply a continuous input signal to the material, as it is in the current case for the designs of deformable mirrors, and so the amount of energy necessary to hold the mirror in a constant position can be significantly reduced.

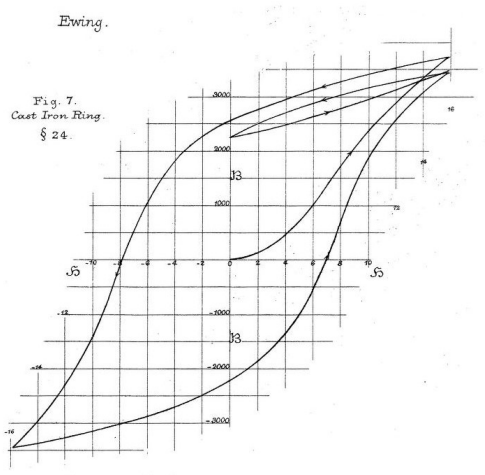


Figure 5 - Hysteresis Loop described by Ewing in his publication.

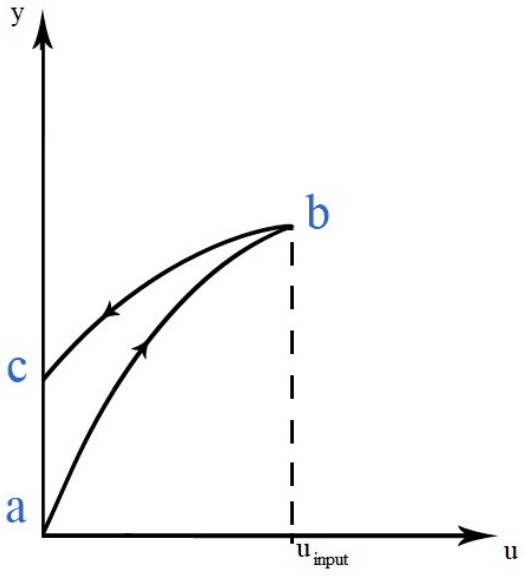


Figure 6 - Schematic of the resulting remnant deformation (c) as the result from applying u_{input} (b) when starting from $0V$ (a).

1.2.3 - Multiplexing

Since there is no need to maintain a constant signal in every actuator due to the remnant deformation, it is not required for every single actuator to have a dedicated control channel, and so several actuators can share the same control channel by doing a time-division control. This is called multiplexing, and gives a very practical solution for the problem most deformable mirror have regarding the complexity of the wiring. If every actuator would need its dedicated channel in a $N \times N$ grid, it would be necessary N^2 dedicated channels, in total. But with multiplexing, and as illustrated in Figure 7, only $2N$ control channels would be required, drastically decreasing not only the complexity of the wiring, but reducing again the amount of energy necessary for the system.

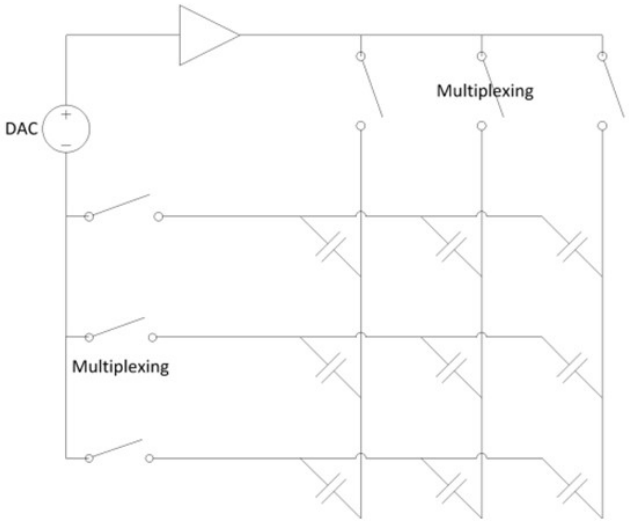


Figure 7 - Example of a system with multiplexing.

1.3 - The proposed HDM

As stated before, this thesis is inserted in HDM project started by the Netherlands Institute for Space Research (SRON). A simple schematic of the cross-section of a standard MEMS DM (Micro-electro-mechanical system deformable mirror) is presented in Figure 8.

The core idea of the proposed DM is to stack layers of actuator strips, where the direction of each layer is orthogonal to the previous one, and so making the intersection points our actuators, and in between each layer of actuators, a film layer of the desired piezoelectric material will be applied. On top of it all, a layer of reflective material would be placed. Figure 9 shows one layer of the piezoelectric film in between two layer of actuator strips. Figure 10 shows a simulation when applying voltage to one of the actuators.

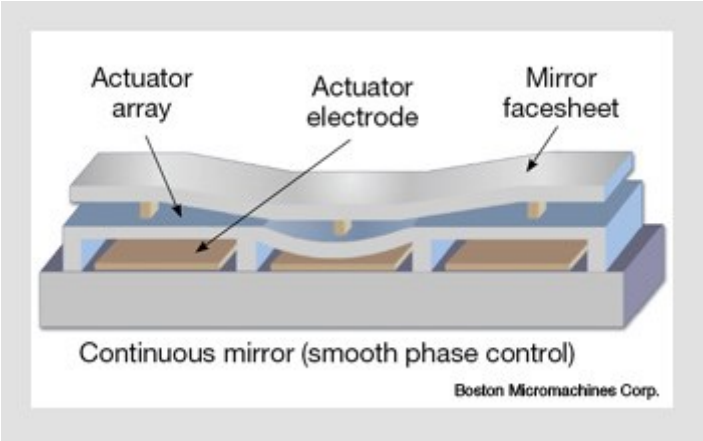


Figure 8 - Cross section of a MEMS DM. When an electric signal is applied in the electrode, the actuator moves, and so will the mirror sheet attached to it.

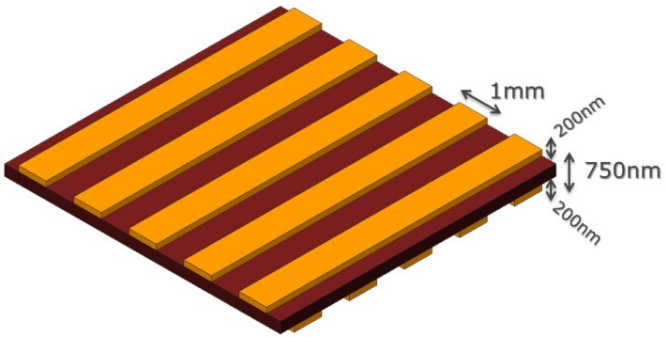


Figure 9 - Schematic for the arrangement of the actuators, with the material film.

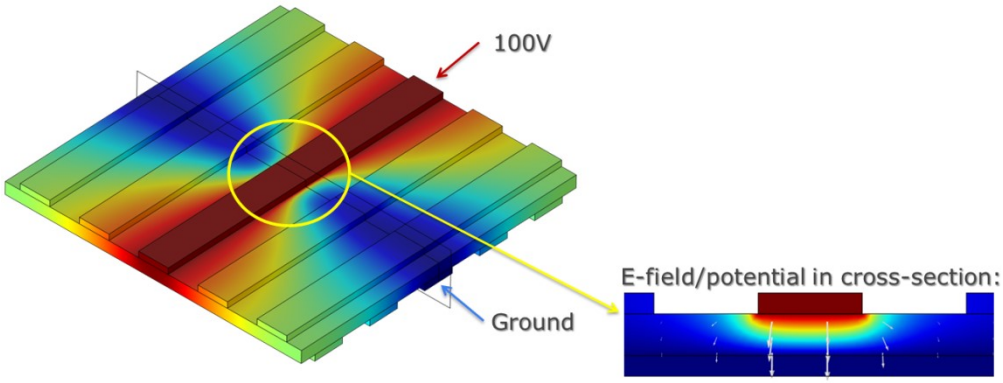


Figure 10 - Result of a simulation made with the Figure 9 Schematic, when applying to one of the top actuators $100V$ and grounding one of the bottom ones. We can see on the right how only the region between the 2 actuators is significantly affected by the electric field

1.4 - Problem definition, research goal, and research questions

Because of the nonlinear behavior of the hysteretic behavior, it is not a simple task to control the deformation of the material. Independently of the design, using piezoelectric materials as the fundamental basis of the actuators of the HDM technology requires a deep understanding of the hysteretic behavior, and how would be possible to control the deformation of the material..

The first task would be to have a model that well describes the hysteretic behavior. This point will be further discussed in Chapter 2. After that, and assuming that a well behaved model is reached, the next step is to find a process to control the remnant deformation of the material. Therefore, the problem statement and research goals for this thesis will be defined as:

Problem statement: *The HDM control system does not have a algorithm to control the remnant deformation of the material.*

Research Goal: *Develop a time efficient algorithm, to control the remnant deformation of a piezoelectric material whose behavior is described by an Operator-based model.*

To achieve the Research goal proposed, there are two main questions that need to be answered. The first one would be:

Question 1: *Is it possible to develop a mathematical method to control the remnant deformation of a piezoelectric material?*

Chapter 3 will be dedicated to evaluate properties of the butterfly loop, and based on such properties, use known mathematical methods to develop an algorithm for the control of the remnant deformation. The follow up question would then be:

Question 2: *How fast is the algorithm when considering a full* 64×64 *matrix of piezoelectric actuators?*

The size proposed is considered a reasonable size for a DM application in an AO system dedicated to exoplanet detection, as it has been referred before. The calculations should take, in maximum, around 2 minutes to do, for this grid size. This will be a "back of the envelope" goal for the computing time, and the sub-question that can follow would be:

Question 2.a: *What other properties of the algorithm chosen can improve the computing time?*

In Chapter 4 the results for the algorithm chosen and the computing times will be first presented, followed by a discuss interpreting these results as a way to answer the research questions.

2 - Preliminaries

To chose how to control the remnant deformation of a piezoelectric material, it must be discussed how can hysteresis loops be modeled. In this chapter, it will be shown how some hysteresis loop are characterized. After that, it will be presented the different types of models that exist, and for the case of the Operator-Based ones, some examples of existing models.

2.1 - Hysteresis loops

In Figure 11, on the left, it is presented a simple non-linear hysteresis loop, where the vertical axis corresponds to the deformation of the material, compared to a initial condition, and the horizontal axis corresponds to the input potential applied to the material. This is not the only case of hysteretic behavior, and the right plot in Figure 11 show the case of a butterfly hysteresis loop. Butterfly loops are important for the study in question, for two main reasons: first, their modeling and control is not as easy as for single loop hysteresis, and in fact, most of the existing models for single-loop hysteresis can't describe a butterfly loop [11]; second, most materials that are used for this type of applications, including the ones that have been developed for the purpose of this project, have shown to have a butterfly hysteretic curves when the deformations needs to be significant.

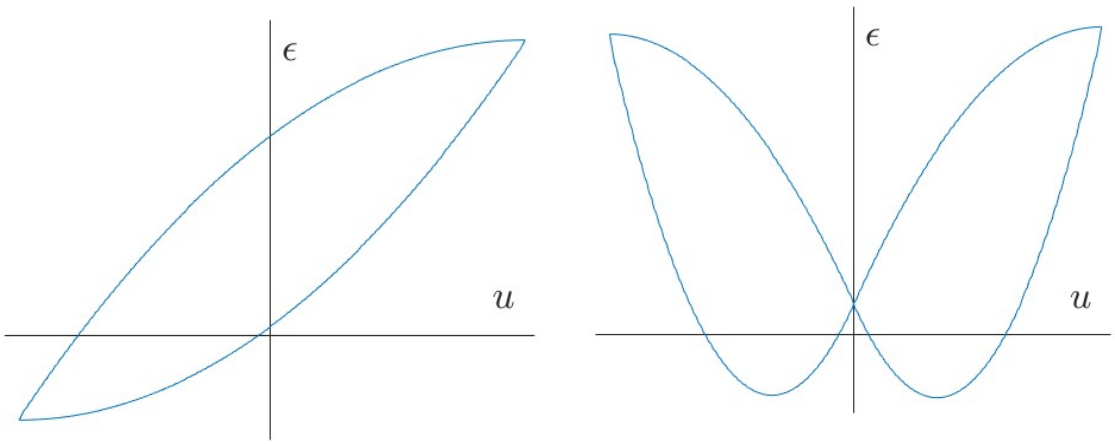


Figure 11 - Examples of the shape of an hysteretic behavior on piezoelectric materials: on the left, the single loop hysteresis; on the right, the butterfly loop hysteresis

2.2 - Physics based models

Physics-Based models try to relate the non-linear behavior of hysteresis with electric and magnetic laws. They consider the physical properties of the material and the electrical and/or mechanical forces applied on it to simulate the hysteretic curve. This help give a direct relation to the origins of the hysteretic performance of the material, which can be important and helpful to understand the hysteretic behavior as a phenomenon. However, this models are too mathematically complex, heavily dependent on the considered system, and as of to date, no physics-based model can explain the butterfly behavior, so these models will not be considered in this work.

2.3 - Differential equation based models

To try to simplify the complexity of the physics-based models, one can try to approach a phenomenological modeling. This means to try to strip away the direct applications of the physical properties of the material in the model, but to try to fit the hysteretic behavior to a simpler numerical model, and then have a indirect relation between the model parameters and the material properties. Models like these often consist of simpler first order differential equations that make a direct relation between the input electric field and the deformation of the material. This approach will simplify the model itself, but make it harder to take a physical meaning out of it.

Again, even though these models are simple to work with, and easier to control, as demonstrated in works made in this work group already [12], to this date no differential model can describe a butterfly loop.

2.4 - Operator based models

Another phenomenological approach is the use of a linear superposition of several relay operators, known as hysterons, each with its own weight, and whose state will depend on the input field applied. These are Operator-Based models. Typically, there will be a weight function, or how will from now on be referred as, a density function, that can be fitted to the considered loop, and the sum of all the relay states with their respective density function value will give the final deformation for the applied input field.

Operator-based models are able to reproduce butterfly hysteretic loops. For this reason, the control method developed in this work will focus on them. We will give examples of some of the most used operator based models in current literature.

2.4.1 - Preisach model

The Preisach model was first introduced by Ferenc Preisach in 1935 [13]. The Preisach operator is the result of integrating a set of infinitesimal relay operators. The relay operator $\mathcal{R}$ can be seen as a simple on/off switch, where the input u will be compared to two values α and $\beta < \alpha$. When u becomes bigger that α, the switch will turn "on", and when it becomes smaller that β, it will turn "off". Mathematically, this can be expressed as:

$$\mathcal{R}_{(\alpha,\beta)}(u(t)) = \begin{cases} 1 & \text{if } u(t) \geq \alpha \\ -1 & \text{if } u(t) \leq \beta \\ \mathcal{R}_{(\alpha,\beta)}(u(t-T)) & \text{if } \beta \leq u(t) \leq \alpha \end{cases}$$

where T is the refresh time (for discrete case, can be seen as sampling time).

A graphical representation of this relay is shown in Figure 12. To obtain the deformation depending on the input potential, the following integral in applied:

$$y\big(u(t)\big) = \iint_{\alpha \geq \beta} \mathcal{R}_{(\alpha,\beta)}\big(u(t)\big)\mu(\alpha,\beta)\, d\alpha d\beta$$

or, in a discrete case:

$$y\big(u(t)\big) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \mathcal{R}_{(\alpha_j,\beta_i)}\big(u(t)\big)\mu\big(\alpha_j,\beta_i\big)$$

where $y\big(u(t)\big)$ is the deformation of the material and $\mu(\alpha,\beta)$ is the density function, giving us a 2D density plane.

The representation of the relay states can be simplified in a plane representation as well, known as Preisach plane. An example of such plane can be seen in Figure 13. This Preisach plane will have two domains, P_- and P_+, where we will have that:

$$\mathcal{R}_{(\alpha,\beta)} = \begin{cases} 1 & (\alpha,\beta) \in P_+ \\ -1 & (\alpha,\beta) \in P_- \end{cases}.$$

The line that divides P_- from P_+, $L(t)$, will change with the evolution of the input $u(t)$, and its where the memory of the material can be seen.

2.4.2 - Prandtl-Ishlinskii model

For the Prandtl-Ishlinskii Model (PI Model from now on), the relay used will have a form of a play operator. The play operator can be mathematically written as:

$$\begin{cases} \mathcal{B}\big(u(t)\big) = \max\left[u(t) - r, \min\big[u(t) + r, y\big(u(t-T)\big)\big]\right] \\ \mathcal{B}\big(u(0)\big) = y_0 \end{cases}$$

where T is the refresh time, and r is the relay's threshold. The final deformation will be given as:

$$y\big(u(t)\big) = \sum_{i=1}^{m} \mathcal{B}(u(t), r_i)\mu(r_i)$$

where $\mu(r)$ is now a density vector. A representation of the play operator can be seen in Figure 14.

For the purpose of a simpler comprehension of the play operator, this can be seen as a form of the relay operator used in the Preisach model, where $\alpha = \beta$, and there is a slope when the switch is about to change. This slope will allow for the switch to be able to have middle states besides being only "on" or "off".

2.4.3 - Krasnosel'skii-Pokrovkii model

The last operator model presented is the Krasnosel'skii-Pokrovkii model (KP model from now on). The relay operator in this case can be observed in Figure 15, and can be calculated from the following expression:

$$KP\big(u(t)\big) = \begin{cases} \max\big(KP\big(u(0)\big), s(u(t), \alpha)\big) & \text{if } \dot{u} > 0 \\ \min\big(KP\big(u(0)\big), s(u(t), \beta)\big) & \text{if } \dot{u} < 0 \\ KP(u(t) - T) & \text{if } \dot{u} = 0 \end{cases}$$

$$s(u, x) = \begin{cases} -1 & \text{if } u \le x \\ -1 + \dfrac{2}{a}(u - x) & \text{if } x < u \le x + a \\ +1 & \text{if } x + a < u \end{cases}$$

where a is a parameter that will be fitted depending on experimental data, and it will define the slope of the operator.

Again, this operator shares a strong resemblance to the one in the Preisach model. In fact, if $a = 0$, the Preisach model is applied. In other words, the KP model is a more general definition of the Preisach and PI models. It shares both the non-reversibility of the PI operator, giving the slope, while having the input boundaries of the switches to happen in non-symmetrical configurations. The result will give a density plane, similar to the case in the Preisach model, but where the Preisach plane values will not be just only 1 or -1, but any value between those.

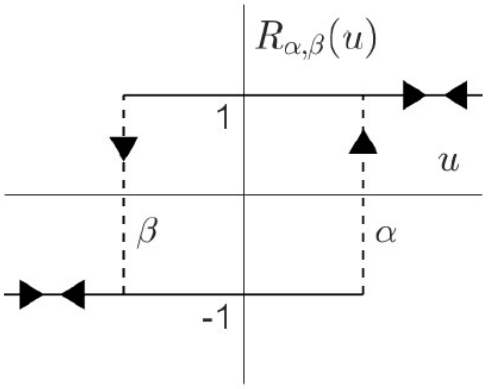


Figure 12 - The Relay Operator

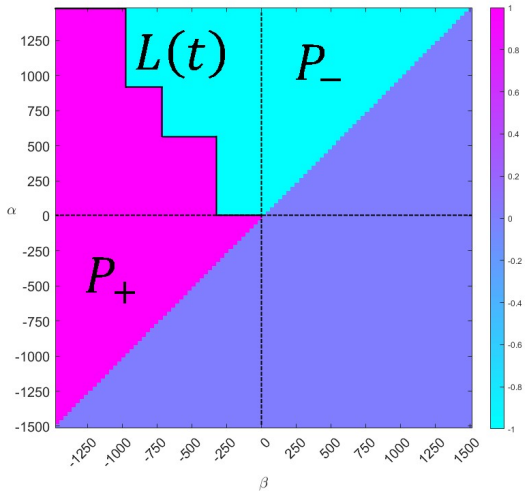


Figure 13 - An example of a Preisach plane

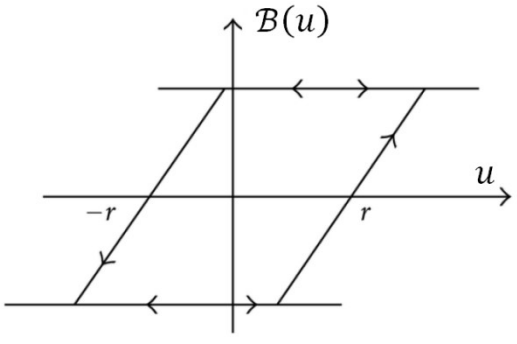


Figure 14 - The PI Play Operator

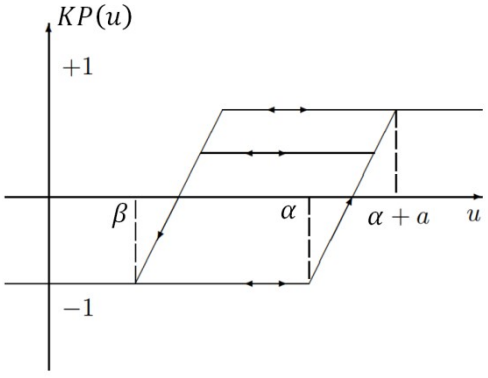


Figure 15 - The KP Operator

[Blank Page]

3 - Method

Only one of the discussed operator-based models will be tested in this thesis: the Presiach model. From previous work done in the research group [14], the Preisach model is the one that best fits the application of controlling a piezoelectric remnant deformation, due to having the smallest amount of error when predicting inner loops behavior, although this error are still significantly high, as it will be discussed later on. It is however important to note that since both the PI and KP Model are somewhat an extension on the Preisach model, the same methods described next can be applied to them as well.

This chapter will be dedicated to reveal the method and algorithm used to control the remnant deformation, based on the chosen model for the butterfly loop.

3.1 - Monotonicity of the remnant deformation

To find a method to control the remnant deformation of the material, it is assumed that it is already given a density plane that well describes the outer loop of its butterfly hysteresis curve.

In Figure 14, we have a butterfly loop, where the zone starting from $0V$ to the operating limits has been divided into 4 sections, where the division points are the local minimums (or maximums, depending on the shape of the loop). Each one of these sections defines a area of the loop where the deformation is monotonic (derivative of the deformation over time maintain its sign) between its limits.

When only operating in one of this sections, the "return" path of the material to its remnant deformation will always be the same (this correspond to the black sections in Figure 16). This means that, when in this returning path, the same relays will be turned off, if going back in the negative direction, or turned on, if going back in the positive direction, independently of the input applied. The remnant deformation will depend on the amount of relays that, when going for the desired input, switched state and cannot be switched back on the returning path. Together with the fact that the deformation in the path going in the direction of the desired input is monotonic, this gives that the remnant deformation will also be a monotonic function of the maximum or minimum input applied.

This is, however, an empirical observation of the butterfly loop behavior, and it can depend on the values of the density plane. Mathematical proof for this should be investigated. For the purpose of this thesis, it will be assumed this behavior is true, and in Appendix A a deeper explanation of this is assumption and the testing made to reach this conclusion is presented.

The monotonic behavior of the remnant deformation in the selected section of the loop was introduced due to the fact of being one of the requirement for the Root Finding method that will be discussed in section 3.3.

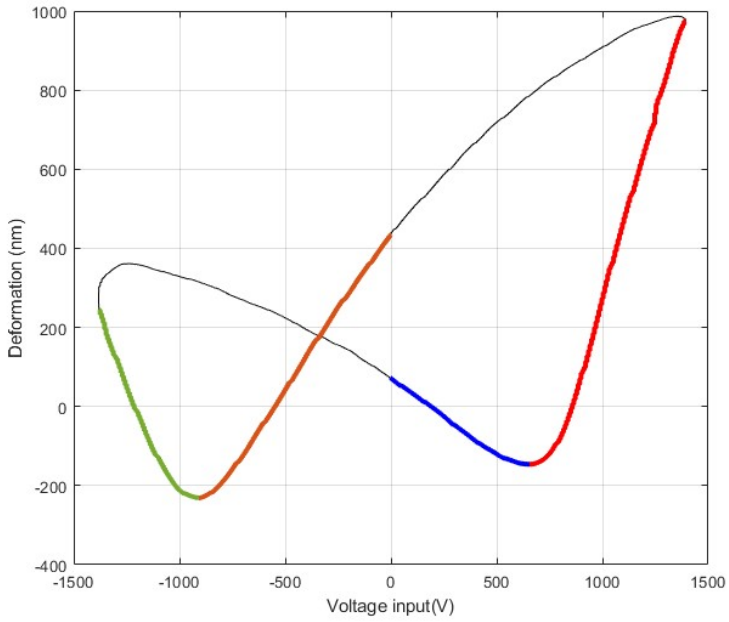


Figure 16 - The 4 evaluated sections for the algorithm.

3.2 - Root Finding problem

Recall the definition for the remnant deformation as a function of the input potential u applied, when starting at $0V$, as $r(u)$. Having r_f as the desired remnant deformation, and having the function:

$$R(u) = r(u) - r_f$$

the goal now becomes to find u_d such that:

$$R(u_d) = r(u_d) - r_f = 0.$$

This can be defined as a Root Finding problem.

There are several methods to solve a Root Finding problem [15], but the only one we will discuss is the Bisection method. Although slower than some other methods like the Newton-Raphson method, these methods depend on also getting the derivative of the deformation, which can be calculated, but at the cost of time, since there is no direct method to have the derivate beforehand and saved like the density plane. The amount of time needed to compute the derivatives would defeat the purpose of its preference. Methods like Van Wijngaarden–Dekker–Brent method do not require computing derivatives and so they can be applied, but they are far more complex that the Bisection for this type of implementation, and for the sake of the timeframe of this thesis, they will not be discussed. The second most simple method to apply would be the Secant method. But, again, given the timeframe of this thesis, only one method could be chosen.

3.3 - Bisection

Let $f(x)$ be a function such that, in a certain interval $x \in [a_1, b_1]$, $f(x)$ is continuous and monotonic. Let's say that $f(a_1) < 0$ and $f(b_1) > 0$. Then, there must be one and only one value z such that $f(z) = 0$.

Let us define $c_1 = (a_1 + b_1)/2$ and evaluate $f(c_1)$. If $f(c_1) > 0$, we now do $a_2 = a_1$ and $b_2 = c_1$. If $f(c_1) < 0$, we do $a_2 = c_1$ and $b_2 = b_1$. The process is repeated until it is reached the condition:

$$|a_n - b_n| < \epsilon$$

where ϵ is a desired threshold for the error when computing $z = c_n$, or, in case of discrete functions, the minimal error possible. Figure 17 show how this process would look on an 1D plot.

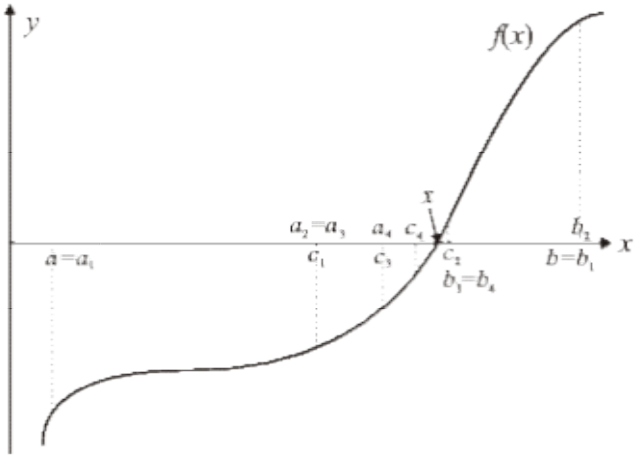


Figure 17 - Graphical example of the steps of the Bisection method

3.4 - Memory resetting

It has been assumed that going only from $0V$, to u_d, and back to $0V$, will give a good enough solution, when that might not be true, or even possible. Because the final deformation of the material depends not only on the final value of input, but also the path it took before, it might be necessary to reset the memory of the material before inputting the desired voltage. Resetting means that the memory of the material is completely wiped by increasing the input voltage to one of its operational extremes. This process takes almost no computing time, allows for the deformation bandwidth to be larger and to achieve the best input electric field for the desired remnant deformation. It is therefore important to consider the case of resetting when testing the optimal applied input for the final remnant deformation.

3.5 - Applying Bisection to the Preisach model: the steps of the algorithm

Given the model chosen, these are now the steps that must be followed to find the maximum input that will give the desired deformation:

1. From r_i, the initial remnant deformation, pick the 2 extreme input values from the section chosen to be a and b
2. Compute $c = (a + b)/2$
3. Calculate $r(c)$ using the chosen model, and compare with r_f
4. Depending on the result of the previous step, change a or b to c, and repeat the steps 2,3 and 4 until an input $c = z = u_d$ is reached.
5. Repeat steps 1 - 4 for each monotonic sections of the loop, and for both cases of resetting or not resetting the material's memory, and compare the eight results to decide which is the best option

In pseudo-code, it can go as follows:

```
u_max       // Maximum electric field input of the section.
u_min       // Minimum electric field input of the section.
e_max       // Maximum admitted error.
final_rd    // Final desired remnant deformation.

def rd=Model(u) // Depending on the chosen model, define the
                   function that returns the remnant deformation
                   when given the desired field input.
a = u_max
b = u_min

rd_a = Model(a)
rd_b = Model(b)

e = abs(a-b)
while e>e_max

      c = (a-b)/2
      rd_c = Model(c)

      // The "if" and "else" conditions might have to be chanced
      around depending on the gradient of the remnant deformation
      as a function of the applied input on the chosen section.
      The code must be able to detect which case to chose from.

      if final_rd > rd_c
            b = c
      else
            a = c
      end
      e = abs(a-b)
end
return c
```

4 - Results and discussion

4.1 - Result for a 64 × 64 grid

This chapter will be dedicated to present the performance result from the algorithm. The algorithm will be ran as if trying to control a 64×64 grid of actuators modeled by the same density plane. Random input conditions (remnant initial deformation and respective Preisach plane) will be applied, as well as random final remnant deformation. Randomizing will assure it is possible to get an overview of all the possible variations in the remnant deformations, within the bandwidth of the material, and that the algorithm is capable of handling any variation of the remnant deformation.

The density plane used to control the data has been acquired following the methods described in previous work in the HDM group [13]. In Table 1 the results from different n × n sizes of the density plane are presented. These results are obtained by making a statistical analysis of the error distributions for the 4096 actuators, for each density plane. Figure 18 shows an histogram of the error distribution for the case of $n = 200$, for the purpose of exemplification. The histograms for each test are presented in Appendix B. The error distribution correspond to the error between the computed remnant deformation after applying the input resulted from running the algorithm, and the desired random deformation. The bandwidth of the remnant deformation was empirically observed from the outer butterfly loop to be 570nm.

The code used to create the algorithm and obtain these results are presented in Appendix C.

n	Standard Deviation (nm)	Standard Deviation (% of bandwidth)	error < 1% (%)	error < 2% (%)	Computing Time (s)
10	80,32	14,09	13,35	25,22	1,57
20	31,72	5,57	25,54	40,45	1,78
25	22,33	3,92	34,23	53,42	1,63
50	10,40	1,83	52,81	74,49	4,45
75	6,20	1,09	69,68	90,09	8,81
100	4,70	0,82	78,78	97,39	16,64
125	3,75	0,66	87,01	99,27	23,97
150	3,22	0,56	90,72	99,76	33,27
200	2,70	0,47	95,26	99,61	58,06
250	2,12	0,37	98,61	99,83	102,69
300	1,74	0,31	99,73	99,85	181,11
400	1,55	0,27	99,56	99,76	621,88
500	1,45	0,26	99,66	99,78	1081,22

Table 1 - The results of the performance of the algorithm for an 64×64 array of actuators

The second and third column represent the standard deviation of the error distributions, in nanometers and in percentage of the remnant deformation bandwidth, respectively. The fourth and fifth columns give the percentage of the actuators that got an error less that 1% and 2% of the remnant deformation bandwidth, respectively. In Figure 18, these correspond to the green and yellow areas of the histogram. The sixth column represents the total time it took for the algorithm to compute the desired inputs for the 4096 actuators.

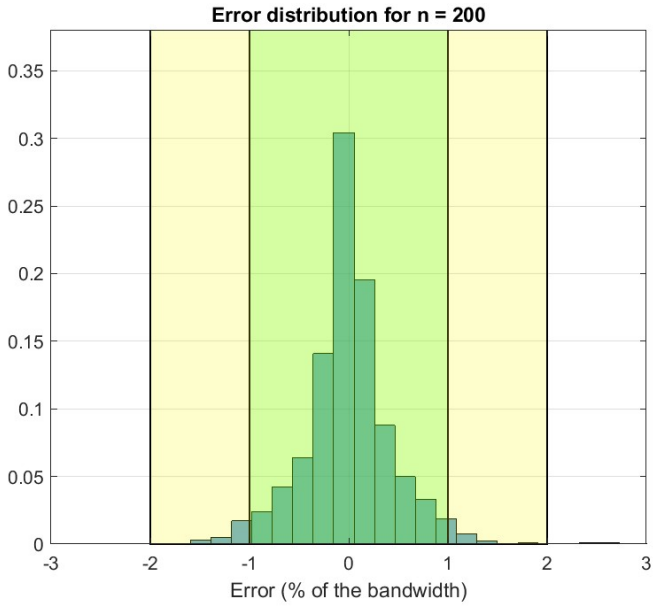


Figure 18 - Error distribution for the case of an 200×200 density plane

An overall look to these results indicates that, for reasonable sizes of the density plane, the algorithm works, and no major errors were detected.

A first interesting conclusion from these results is that the errors in the final computed remnant deformation do not seem to come from the algorithm. In fact, from theory, the algorithm will find the best possible solution for the constrains we indicated, so these error should only depend on the model chosen to describe the hysteretic behavior. However, as discussed in Appendix A, the monotonic behavior of the remnant is not completely guaranteed, and so errors might occur from this assumption. However, from the obtained results, if these errors exist, they seem to be irrelevant compared to the ones caused by the model.

If the algorithm has no influence in the error, the only important factor for the consideration of the algorithm is its speed, to which it can be concluded that the algorithm performs very satisfactorily. Given the considered results, for $n = 300$ there is still a reasonable runtime of around 3 minutes. It is also important to notice that these tests were run into a home laptop, which computing power is far from the one on the hardware that would be implemented in an actual space mission, so bigger sizes of the density matrix could potentially be considered for these timeframes, and that Matlab© might not be the fasted programming tool to run such methods.

The columns for the errors bellow 1% and 2% of the bandwidth give important information about the minimal sizes of the density matrix, given the chosen model. With $n = 125$, the percentage of actuators with an error bellow 2% is above 99% for the first time, but the errors under 1% is still below 90%. But at $n = 150$, it becomes above 90%, so this would be the minimal size for an ideal density plane. With this size, the computing time for the algorithm is around 30 seconds.

4.2 - Problems with the Preisach Model prediction

Despite the obtained result showing a very promising process to control the remnant deformation of a piezoelectric actuator modeled by a Preisach model, these results do not take into account the bad predictive behavior of the Preisach model for inner loops [16]. This would increase the errors significantly. The same problem will also rise from the other discussed models.

Without considering the use of a different model that would predict the behavior for inner loops better that the Preisach model, one solution to solve the problem with its predictions would be to run the algorithm in an "open loop", where for every calculation we did for a single actuator, a new density plane would be fitted, with operational input limits just a fraction bigger that the input result in the previous itineration. One of the problems with this approach would be the multiplicative effect it would have on the computing times, to the point where some of the sizes for the density planes could no longer be considered. The other big problem is the computing of the density plane, which not only also requires much longer to compute, so adding even more computing time, but would also require to have some type of system where we could obtain the inner loop without actually operate the actuator, so the initial conditions would not have to be reproduced again, or be forced to reset the memory of the operator every time the algorithm had to be ran.

One other solution would be to just have an arbitrary number of density planes, for different value of maximum operational limits, and when running the algorithm, test only to the one that has a remnant deformation of the outer loop closest to the desired output. The advantage of this is that the system could work in a closed loop, avoiding the computing time to obtain new density plane, since this would be already saved in memory, but the prediction error would still exist, but just at a smaller scale. The biggest disadvantage is that the initial condition would have to be carefully and precisely chosen in order to make sure they would correspond to the right loop, and that would also require to operate the actuator so he would be in the proper initial state, and that would cost energy.

Testing how the predictions of the Preisach plane would influence the result from the algorithm created in an actual physical setup of the system is a very important step for this project. Unfortunately, no experiment setup was available to perform this tests.

4.3 - Improvements to the algorithm

If the implemented algorithm has to be further improved for smaller computing times, this section will be dedicated to propose some modifications of the algorithm with which it would be possible to obtain better computing times.

- Root Finding Method

As briefly discussed in Section 3.2, there are other Root Finding Methods that could potentially be used for the purpose of this thesis. The chosen method, Bisection, is one of the slowest ones. Choosing any other methods will have advantages when it comes to computing time, but the complexity and requirement to the use of these methods can prevent them from being better choices. The best candidates would likely be the Secant method and Van Wijngaarden–Dekker–Brent method, since these do not require to compute derivatives of the remnant deformation.

- Reducing the number of tests for a single actuator

As it is, for every single computation for one actuator, the algorithm performs 8 tests: depending if it needs to go to the positive inputs or the negative inputs, which section in that input direction would be tested, and if resetting the material is necessary. Currently, the algorithm makes a brute force test to each combination of these conditions, and returns the one with the smallest absolute error. If it would be possible to find that for a given computation some of these conditions would not give the desired result before computing it, this would reduce the number of test for a single computation significantly. For example, knowing if the material needs resetting or not before testing or if we need to go to the positive or negative directions of the input could potentially reduce the computing time by half. The problem becomes to identify these cases. Further study of these condition could be made to make sure brute force of all conditions is not necessary, and that way improve the computing times.

- Preisach's planes reduction

An interesting observation from the Preisach plane with that the boundary $L(t)$ always evolves in the negative direction of α, and the positive of β, as illustrated in Figure 19. Also, since it is being considered the remnant deformation at $0V$, the boundary $L(t)$ will always have the finishing point at $(\alpha, \beta) = (0,0)$. These two conditions give that any relays with $\alpha < 0$ will belong to P_+, and the ones with $\beta > 0$ will belong to P_-, for all the cases. This means that only the area where $\alpha \geq 0$ and $\beta \leq 0$ need to be evaluated for the control, since the rest of the Preisach plane will remain constant. This can be very useful to make the control algorithm be faster by significantly reducing the size of the computed matrixes. However, and despite the fact that would be a small adjustment to the implemented code, this was not used on the control algorithm presented in this thesis, only because the code written was simplified in order to easily adapt it to be used for other operator based models, to which this "trick" cannot be applied. But in an application where the Preisach plane is considered, this property should be taken into account to improve the algorithm's performance.

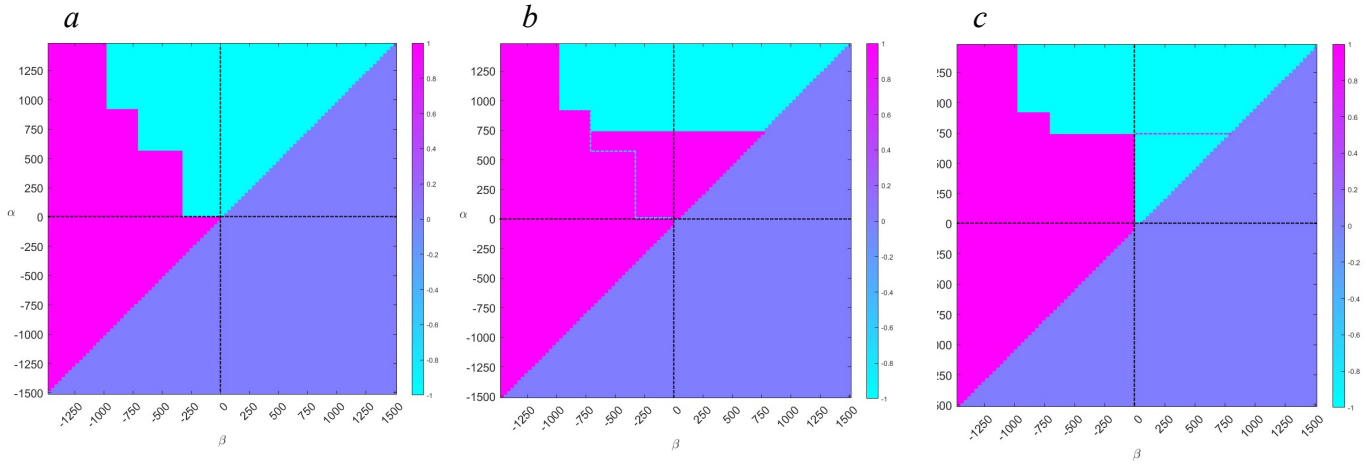


Figure 19 - Example of the evolution of an Preisach plane with the applied input, from *a* (at 0V) to *b* (to a maximum applied input) and to *c* (back to 0V).

- Resetting the material's memory for every computation

One simple implementation that would reduce the computing times significantly would be to force a reset on the material for every single new calculation. This would force for the initial condition to be always the same, and so it would be possible to store in memory the function of the remnant deformation as depending on the applied maximum input, saving significant computing time. This would also allow to compute and store the derivative of it, allowing to use other Root Finding methods like Newton method without having to make time consuming calculations for the derivatives. The biggest disadvantage of this would be the operational cost: resetting every single time not only would require more power to be spent for every single time a new remnant deformation had to be achieved, but it could also potentially degrade the HDM much faster, since the physical stress applied on it would be much larger. This is therefore related to the durability of the HDM, and intensive physical test for it had to be conducted to know if resetting the material for every itineration is possible.

[Blank Page]

5 - Conclusion

The works on this thesis proposed to start the research on how the remnant deformation on a piezoelectric material could be controlled, assuming we would be dealing with a butterfly loop, have been reached. The main goal, which would be to try to develop a control algorithm to such system, no matter what the performance would be, was achieved. Following that, it could be concluded that, even for the primary and most rudimentary form of the algorithm, as it is now, still shows promising result regarding its performance, meaning that with some of the already proposed modifications to the algorithm, its performance would be even better. However, this results do not take into account the bad prediction of some of the models when modeling inner loop of the displacement loop of the material, and some of the proposed solutions can increase the computing times significantly.

The follow up works on this subject, based on the results of this thesis, should be focused first on applying the algorithm to an actual actuator and evaluate the results. More problems could arrive from this observations that are impossible to predict only by simulating the chosen hysteretic model.

Following the test on actual hardware, the next step should not be focused to improving the algorithm just yet, but to first find a better model to describe the hysteretic behavior of the material. As stated before, the algorithm already finds the best solution for the chosen model, and since the time performance is already relatively good, the biggest problem is the errors that arise from the predictions of the model. Only after the modeling of the hysteretic behavior has been well predicted, the performance of the algorithm should be improved based on some of the propositions made in Section 4.3.

[Blank Page]

Appendix A - Monotonicity of the Remnant deformation

Using the Bisection method requiers the function being evaluated to be monotonic. Proving that the remnant deformation is in fact monotonic as a function of the maximum applied input, for any initial condition of the Preisach plane, shown to be a challenge. It was not possible to find a mathematical proof of such behavior. However, this chapter will provide an explanation based on empirical observations of the model to show the remnant deformation has a prominent monotonic behavior.

A test to observe the gradient of the remnant deformation as a function of the maximum applied input is implemented. Starting from a random initial condition of the Preisach plane at $0V$, the variation of the remnant deformation depending on the maximum input is registered. With it, the derivative of the remnant deformation is taken using the Euler's method. The maximum and minimum value of the derivatives are registered. If, for the same section, the sign of these maximum and minimum derivatives do not change, then we have monotonic behavior. This was repeated 5000 times, for two sections of the butterfly loop: from $0V$ to the input that gives the local minimum of the deformation in the positive direction of the input, and from such same input to the maximum operational input of the loop. An histogram with the result is presented in Figure A.1. The density matrix used for this test is 200×200.

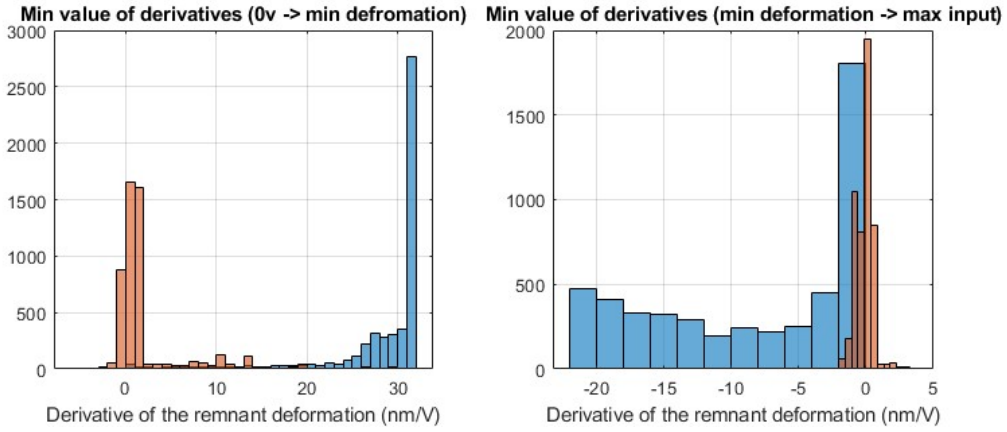


Figure A.1 - The distributions for the maximum values (in blue) and the minimum values (in red) of the derivatives of the remnant deformation.

It can be seen that, for the same section, there were points where the derivative changes sign, which would refute the existence of the monotonic behavior. However, is possible to notice that the variations to the opposite side, represented by the red graphs in Figure A.1 are much smaller in the order of magnitude of the maximum values, in blue. The bandwidth of the derivative around the turning point is itself much smaller than the average value of the derivatives. Therefore, it is reasonable to neglect these small variations, and to consider the remnant deformation has a monotonic behavior as a function of the maximum applied input, for the purpose of this thesis.

Furthermore, these small variation might be the result of approximation errors, such as inaccurate calculations of the derivative, where the Euler methods is not precise enough, or due to the fact the density function used has fitting errors itself.

These observations do not prove that the remnant deformation always has a monotonic behavior, but it can be clearly seen that in most of the times, it is the main behavior. In fact, when running the control algorithm based on the bisection, no problems detecting the zero point due to non-monotonic behavior were ever found.

But let us assume that these small variations would indeed result in errors on the algorithm. The remnant deformation could not be assumed to be monotonic for any initial condition. The easiest solution for this would be to find a initial condition, for a certain section (as referred in Section 3.1, in Figure 16), that would guarantee a monotonic behavior. One obvious initial condition to consider would be to just reset the material's memory. Testing for a 200×200 density matrix, resetting shows to give a monotonic behavior for all the four section of the loop, for the considered density plane, as it can be seen in Figure A.2. This would require for the material to be reset every time a new remnant deformation would be needed. In chapter 4, the advantages and disadvantages of this constant resetting have been presented.

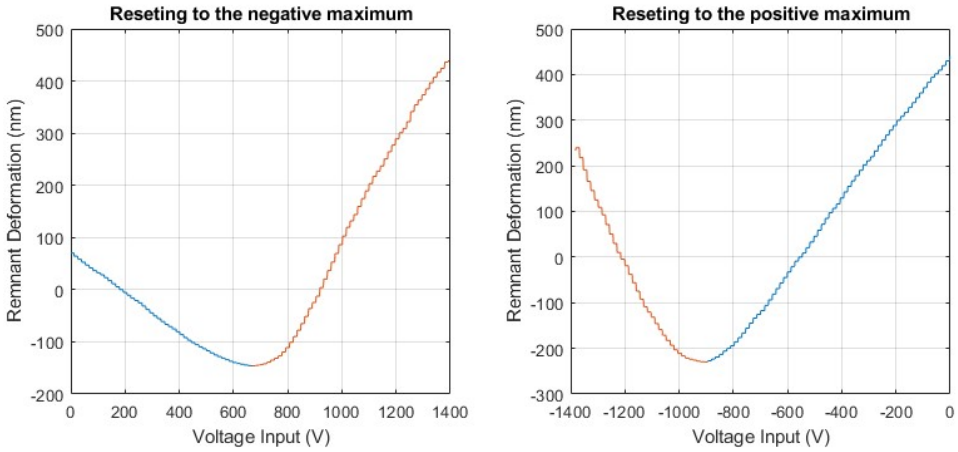
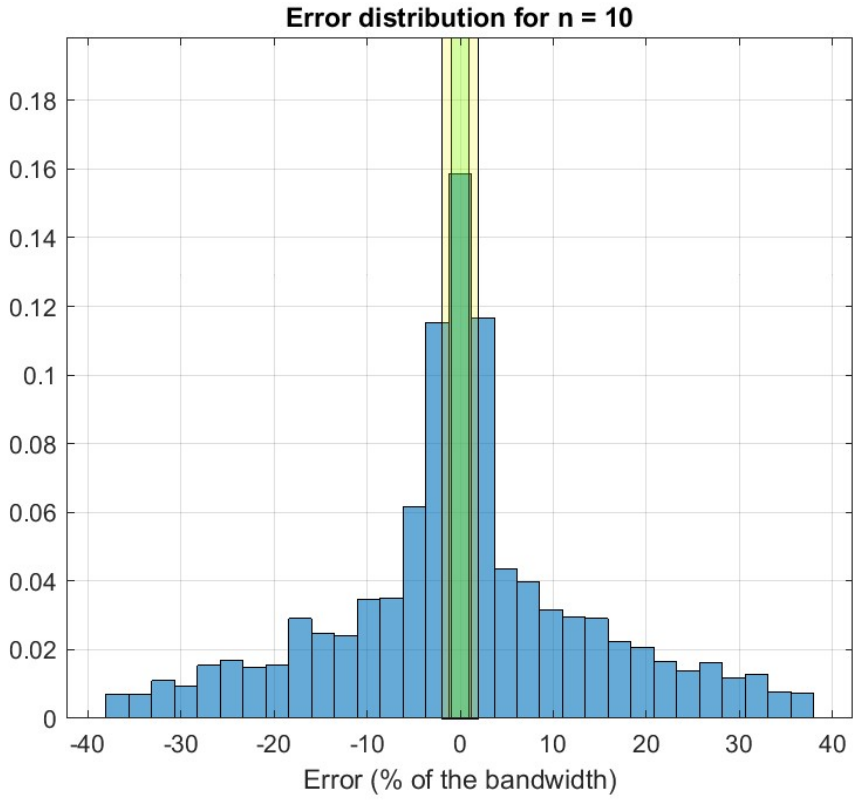
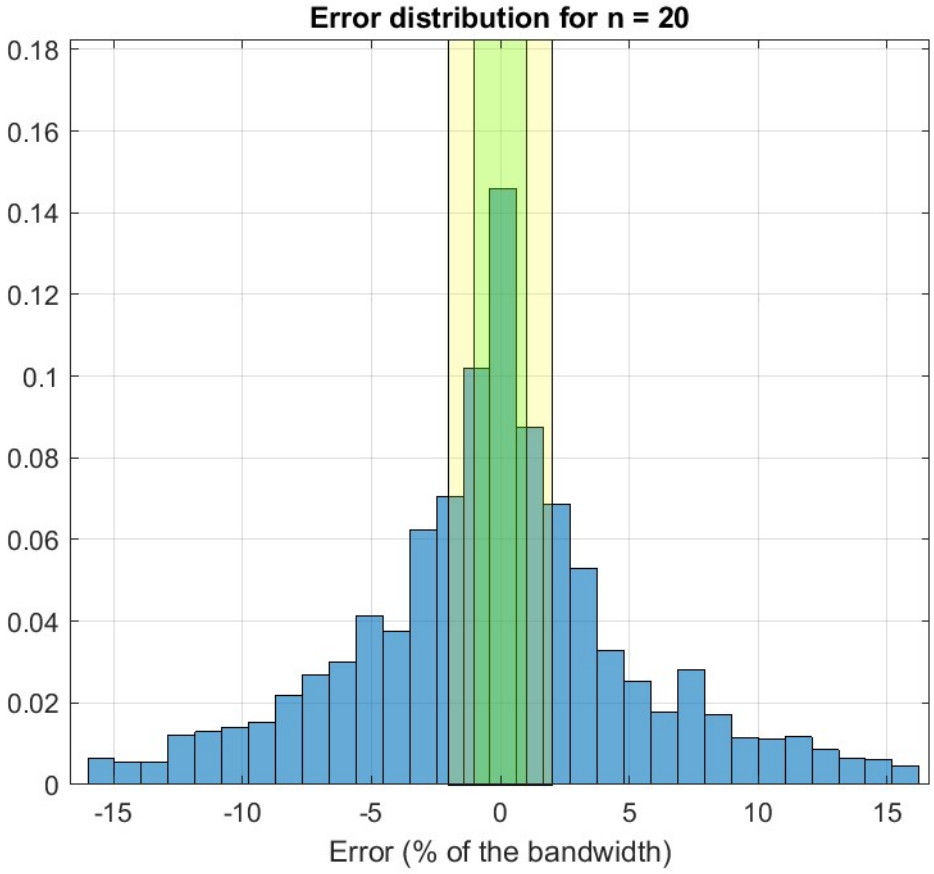
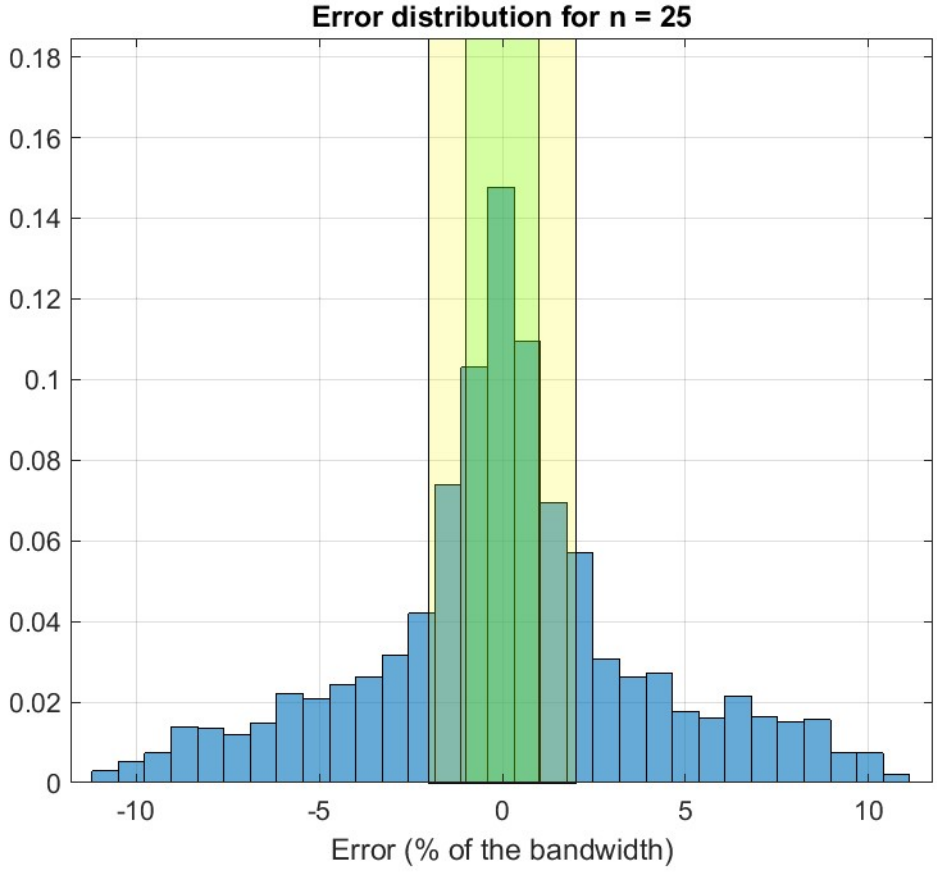


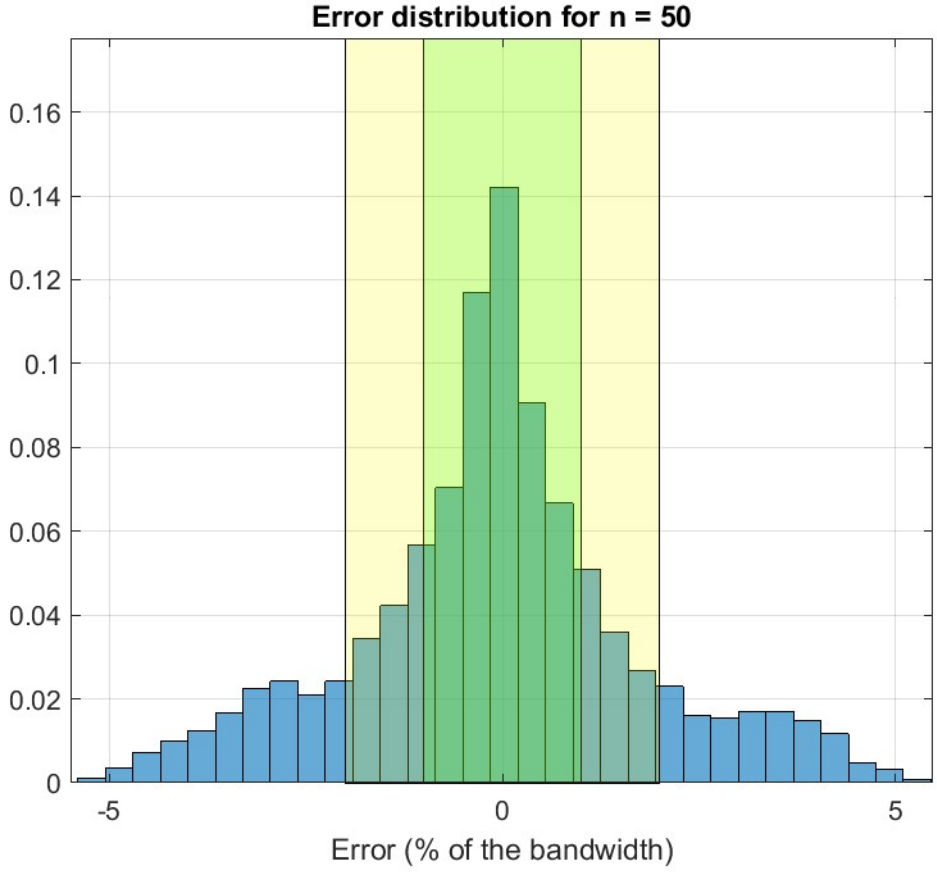
Figure A.2 - Remnant deformation as a function of the maximum applied input, when the memory of the material is reset to the negative operational maximum (left plot) and to positive operational maximum (right side). Again, each plot divided into the sectors that show a monotonic behavior (red and blue in each graph).

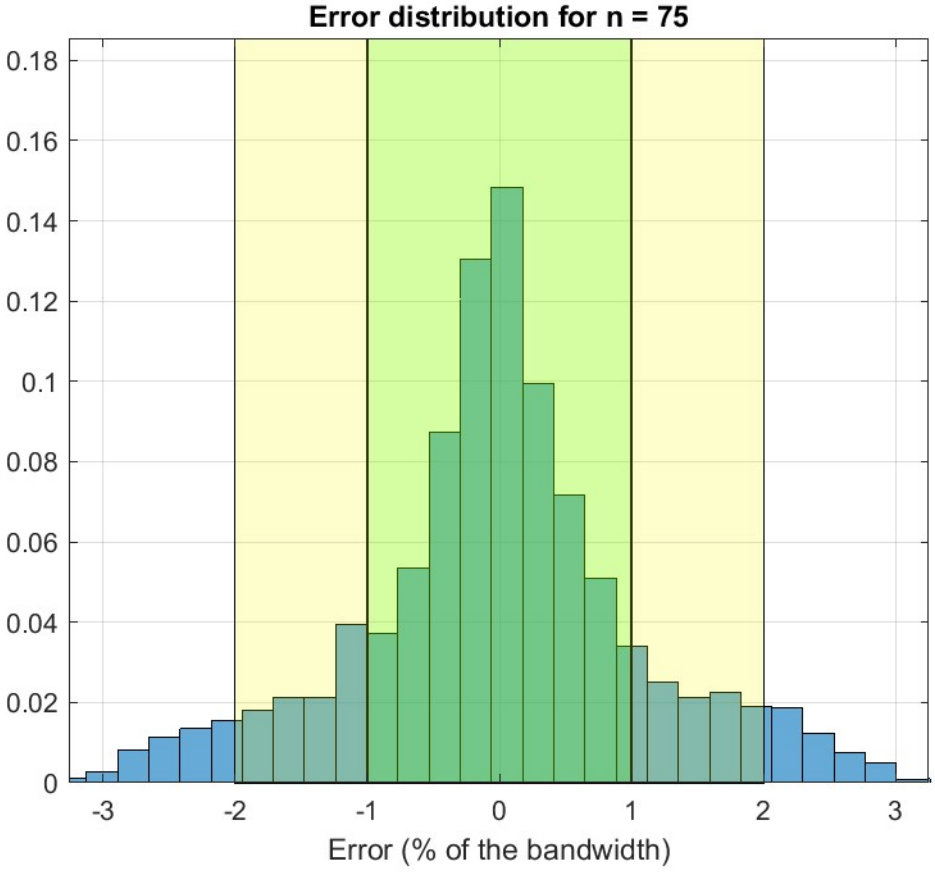
Appendix B - Error distributions

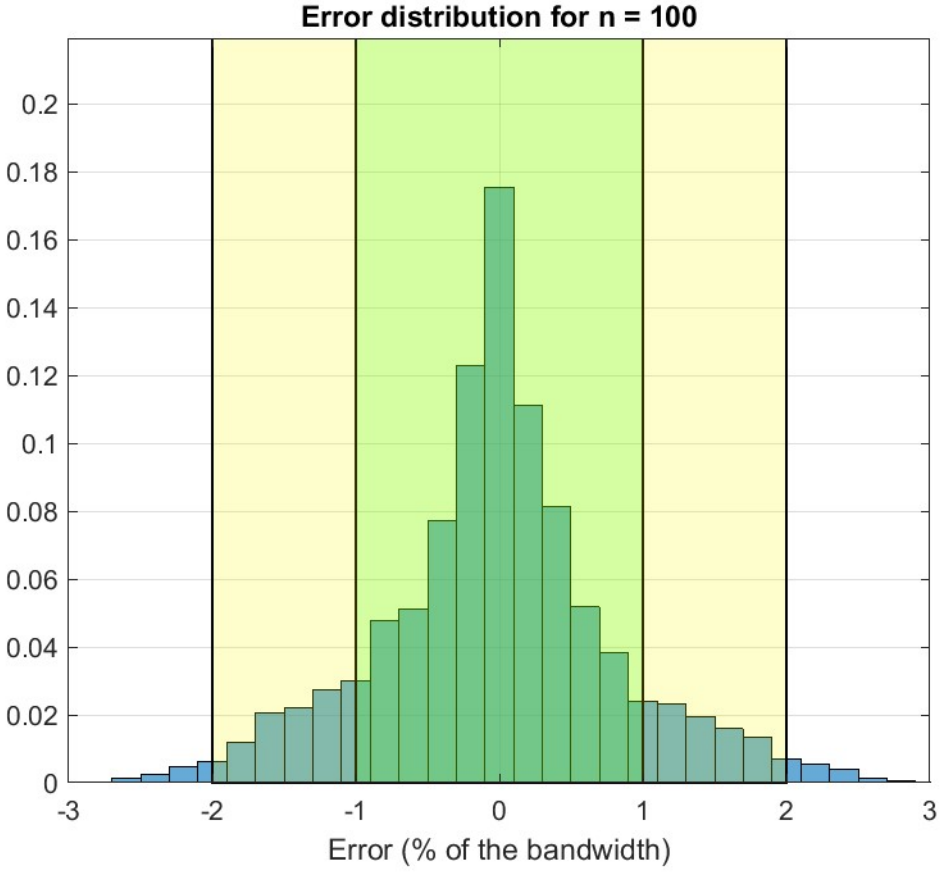


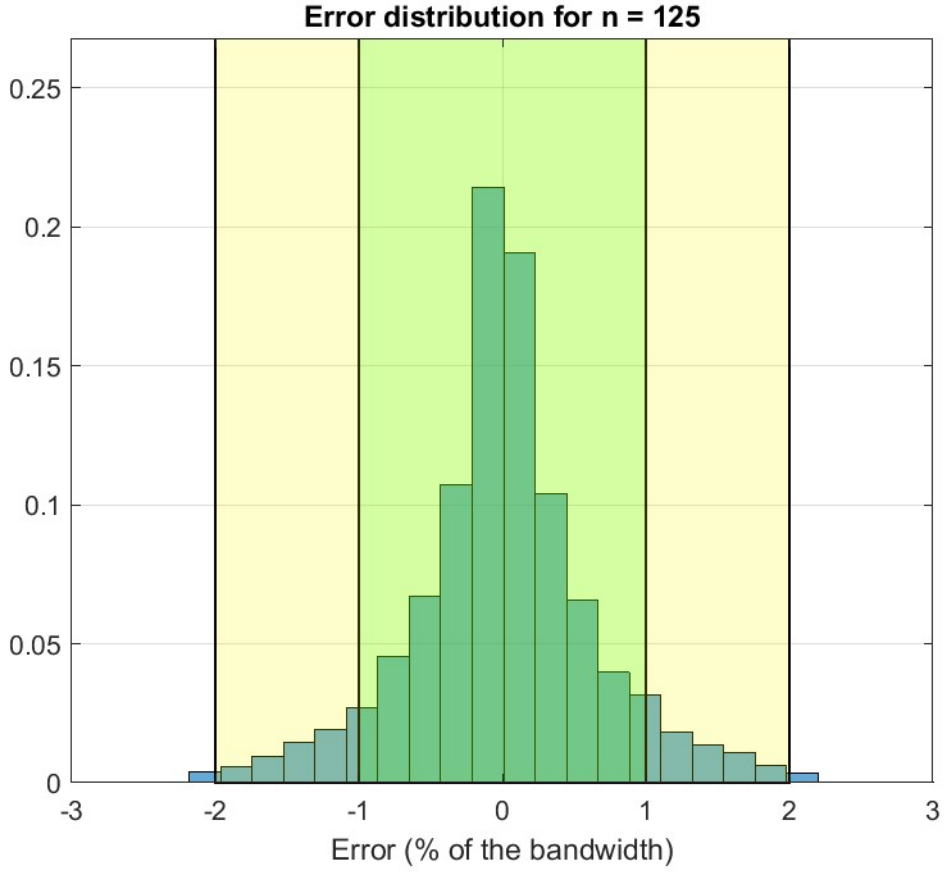


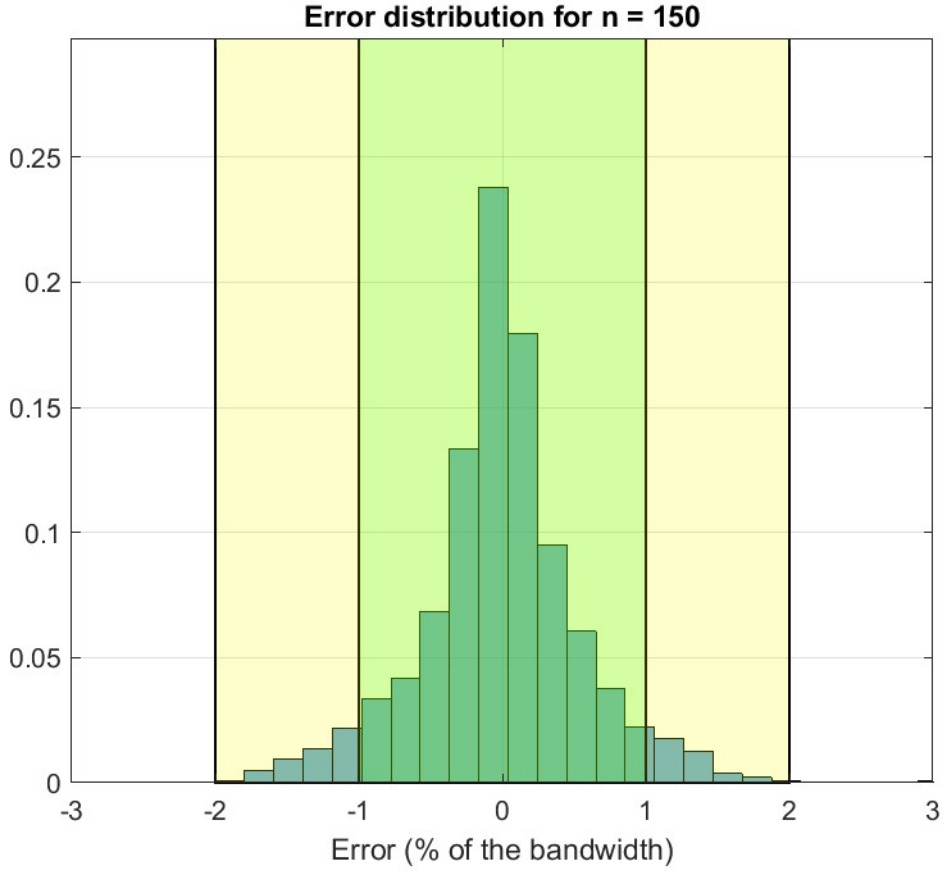

Error distribution for n = 25
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-10
-5
0
5
10
Error (% of the bandwidth)

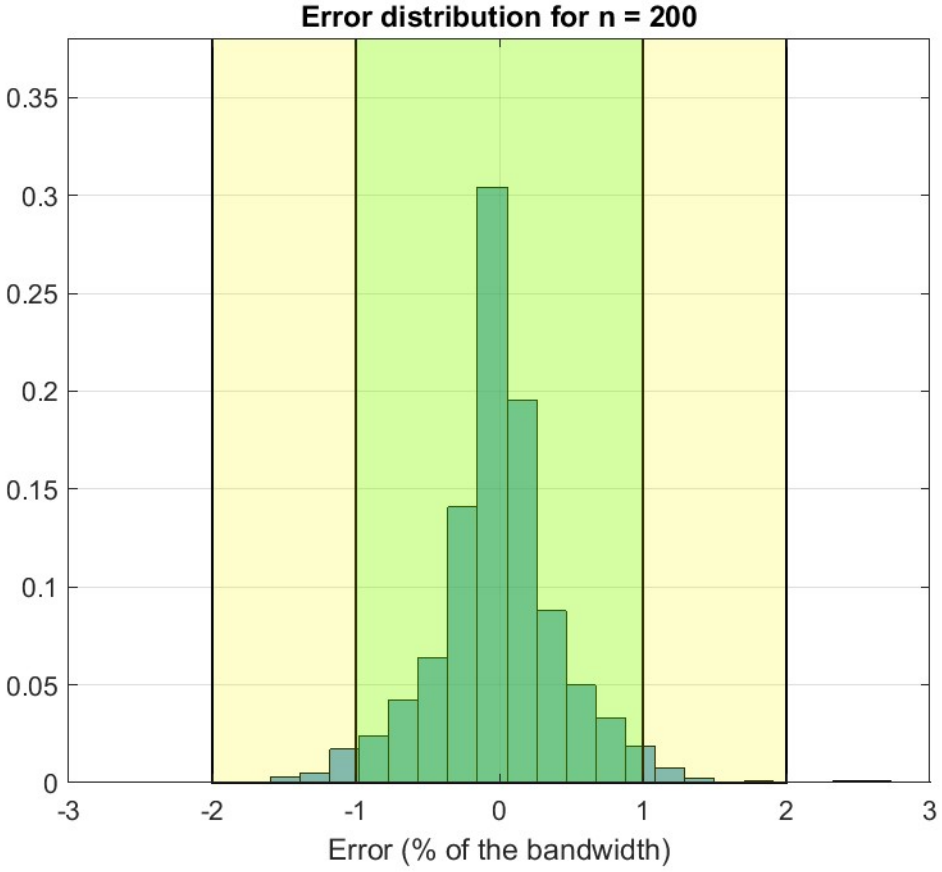

Error distribution for n = 50
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-5
0
5
Error (% of the bandwidth)

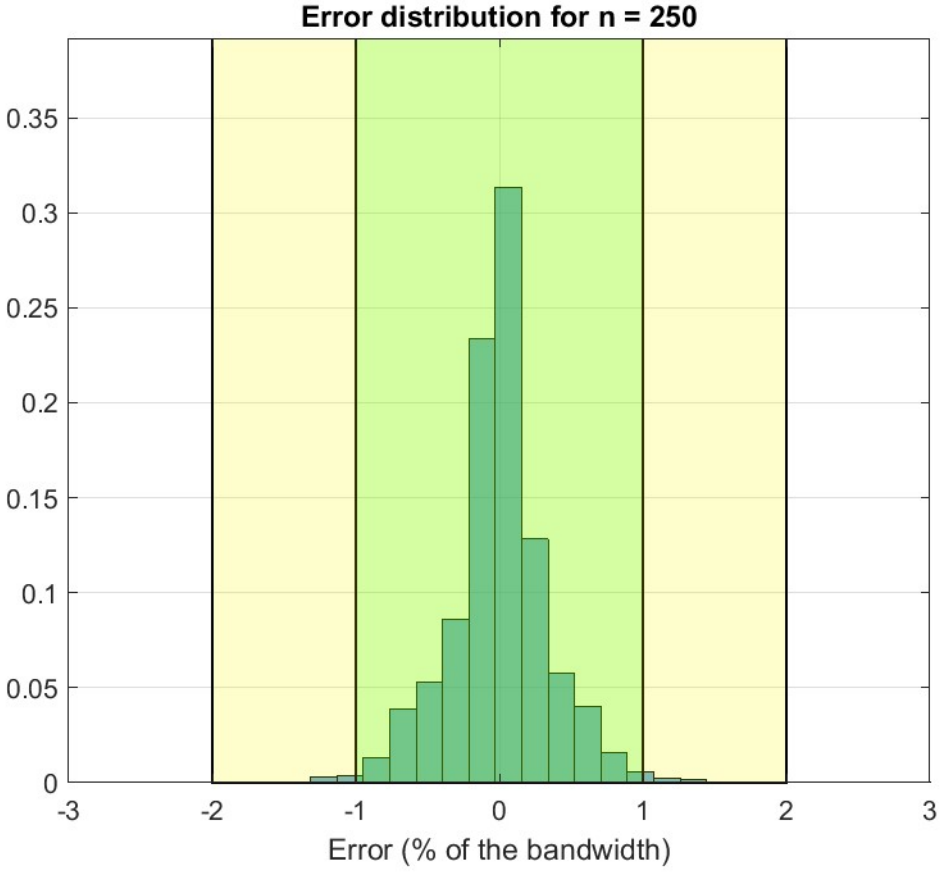

Error distribution for n = 75
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)

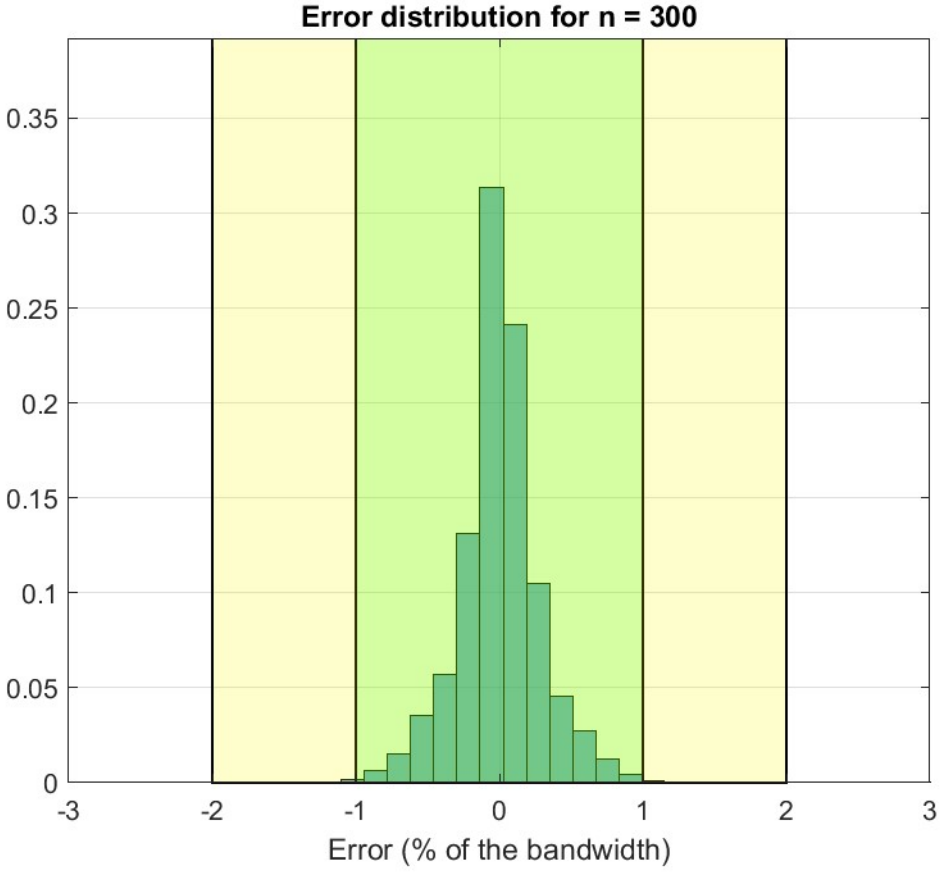

Error distribution for n = 100
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)

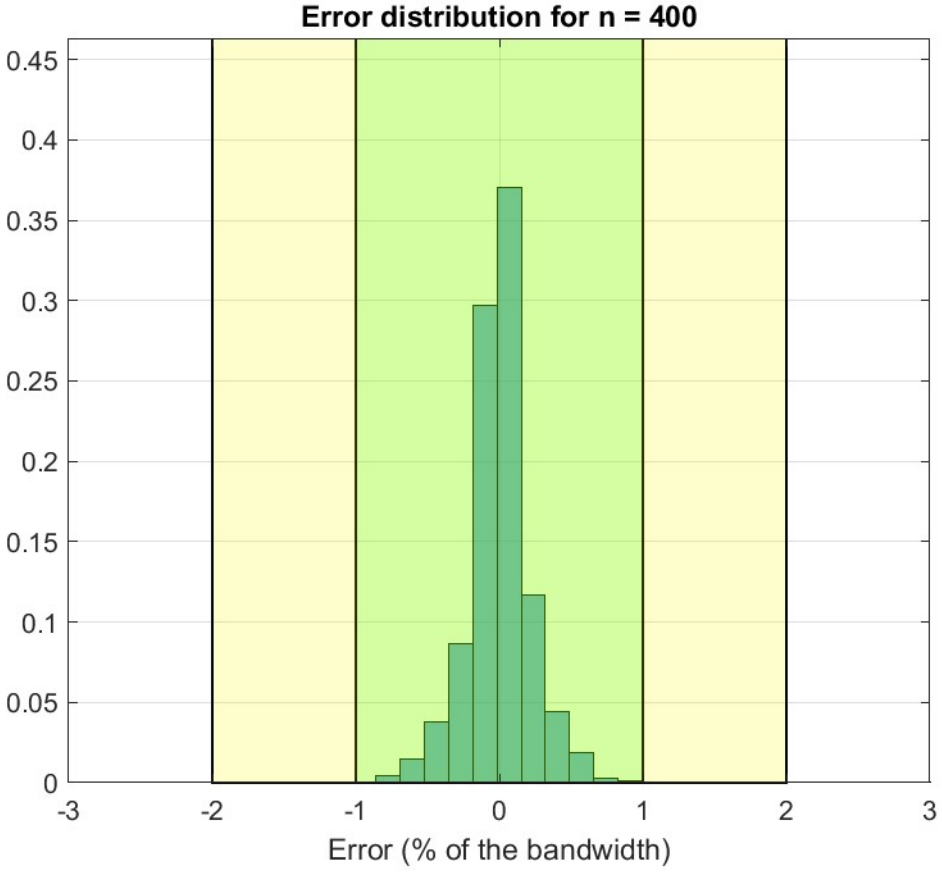

Error distribution for n = 125
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)

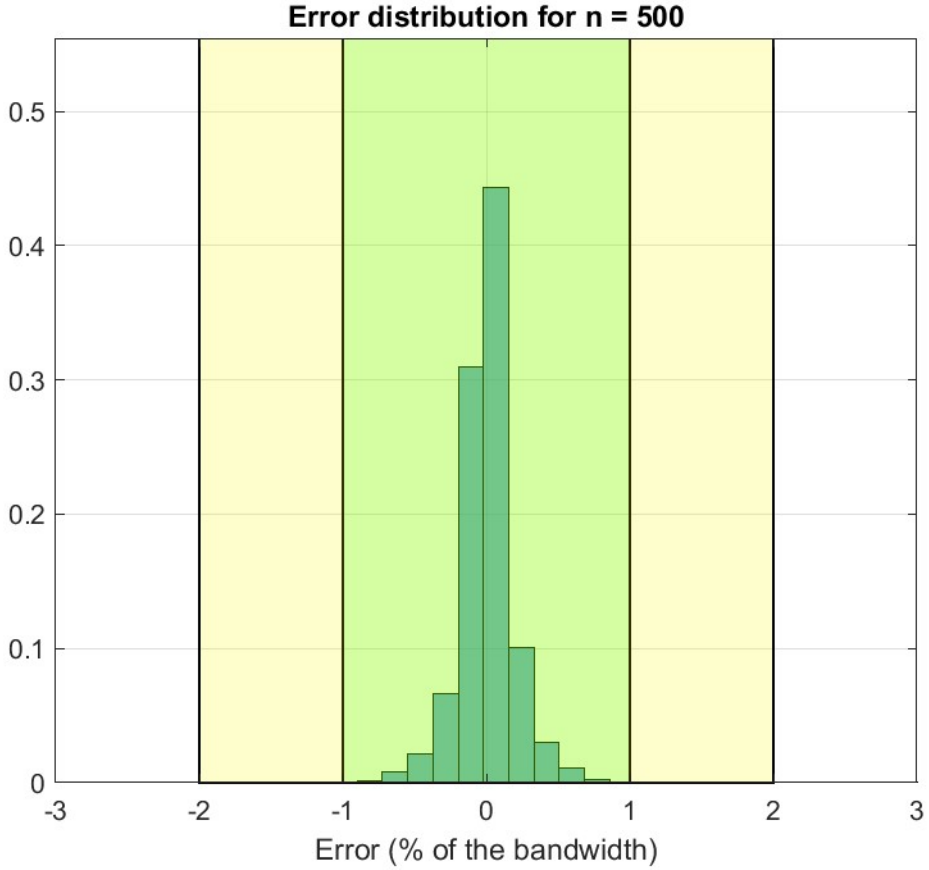

Error distribution for n = 150
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)


Error distribution for n = 200
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)


Error distribution for n = 250
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)


Error distribution for n = 300
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)


Error distribution for n = 400
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)


Error distribution for n = 500
0.5
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
3
Error (% of the bandwidth)

[Blank Page]

Appendix C - Codes

The codes created in Matlab© to achieve the results published in this thesis are here presented.

C.1 - Pseudo-random Preisach plane

```
function P = Random_Preisach_Plane(alpha, beta)
    % Generates a pseudo-random Preisach Plane
    %
    % INPUTS:
    %
    % alpha - alpha column vector
    % beta  - beta row vector
    %
    % OUTPUTS:
    % P - Random Preisach Plane

    N_alpha = length(alpha);

    % Random number of turning points
    R = 9;  % Changes number of turning points
    n = round(R*rand()) + 1;
    flag = round(rand());

    % Random turning points

    rand_u = [];
    for i = 1:n
        if rem(flag+i,2) == 0
        rand_u(i) = alpha(1+round(0.5*(N_alpha-1)*rand()));
        else
        rand_u(i) = alpha(round(0.5*N_alpha*(1+rand())));
        end
    end
    P = -1.*sign(rand_u(1)).*ones(N_alpha);
    for j = 1:length(rand_u)
        u = rand_u(j);
        if u>0
            P(alpha<=u,:) = 1;
        else
            P(:,beta>=u) = -1;
        end
    end

    P(alpha<=0,:) = 1;
    P(:,beta>=0) = -1;

    Z = triu(ones(size(P)));
    Z = rot90(Z);

    P = P.*Z;
end
```

C.2 - Evolving the Preisach plane

```
function P = Evolve_P(P,input,index,direction)
    % Calculates the final state of the Preisach plane when going from
    % the initial state to a desired input
    %
    % The goal of having this as a separate function is so that its
    % possible to easily change to anotehr model
    %
    % INPUTS:
    %
    % P         - initial state of the Preisach plane
    % input     - index input to which we desire to go
    % index     - alpha and beta index vector
    % direction - defines if we are moving to the desired input
towards
    %             the maximum input (increasing inpput, then back to
0)
    %             or minimum input (decreasing input)
    %
    % OUTPUTS:
    %
    % P - Final state of the Preisach plane

    if direction == 0
        P(index>=input,:) = 1;
    else
        P(:,index>=input) = -1;
    end

end
```

C.3 - Finding the maximum applied input using Bisection

```
function [input, P_final, yf_calc, error, time, steps_check, steps_ab,
reset_flag] = Find_max_input(density_plane, alpha, beta, P, yf,
steps_check, steps_ab)

    % Compute the input necessary to achieve desire remnant
deformation
    %
    % INPUTS:
    %
    % density_plane     - the denisty plane that models our actuator
    % alpha             - alpha column vector
    % beta              - beta row vector
    % P                 - initial Preisach plane
    % yf                - desired final remnant deformation
    % steps_check       - flags if the local minimums of the butterfly
loop
    %                   have already been found (0 if not, 1 if yes)
    % steps_ab          - returns the index in alpha and beta vectors
that
    %                   limit the sections the algorithm runs. It's
only
    %                   computed 1 time, if steps_check == 0
    %
    % OUTPUTS:
    %
    % input             - the voltage input needed to be applied to
    %                   achieve the desired remnant deformation
    % P_final           - final Preisach plane (for debugging)
    % yf_calc           - the computed final remnant deformation,
according
    %                   to the chosen model and computed input
    % error             - difference between the desired and the
computed
    %                   final remnant deformation
    % time              - time to compute the input necessary
    % steps_check       - (check INPUTS section)
    % steps_ab          - (check INPUTS section)
    % reset_flag        - Indicates if the result input voltage is
applied
    %                   after reseting the material

    % Defining alpha and beta's Zeros, and Z
==============================

    ab_index = 1:length(alpha);

    alpha_zero_find = alpha;
    alpha_zero_find(alpha_zero_find<0) = max(alpha_zero_find);
    alpha_zero = find(alpha_zero_find == min(alpha_zero_find));

    beta_zero_find = beta;
    beta_zero_find(beta_zero_find<0) = max(beta_zero_find);
    beta_zero = find(beta_zero_find == min(beta_zero_find));

    ab_zero = [alpha_zero, beta_zero];  % saves the index of the zeros
in
                                        % alpha and beta vectors

    Z = triu(ones(size(density_plane)));
```

```
    Z = rot90(Z);

    % Finding Max Input
==================================================

    tstart = tic;    % clocks the time for the calculation
    results = {};    % list to compare different input, and wich is the
best
    for reset = 0:1
        % reset the memory of the material, making it go to one of the
        % input limits
        for direction = 0:1
            % direction defines if we are moving to the desired input
            % towards the maximum input (increasing inpput, then back
to 0)
            %  or minimum input (decreasing input)

            P_temp = P;

            if reset == 1 && direction == 0
                P_temp = Evolve_P(P_temp,1,ab_index,1-direction).*Z;
                P_temp = Evolve_P(P_temp,ab_zero(2-
direction),ab_index,direction).*Z;
            elseif reset == 1 && direction == 1
                P_temp = Evolve_P(P_temp,max(ab_index),ab_index,1-
direction).*Z;
                P_temp = Evolve_P(P_temp,ab_zero(2-
direction),ab_index,direction).*Z;
            end

            % Calculating local minimuns
            if steps_check == 0
                % calculates it once, and passes it on to the next
                % iteneration of the loop, to save time.
                steps_check = 1;
                [steps_ab(1,:),steps_ab(2,:)] =
Find_steps(density_plane, alpha, beta,ab_index,ab_zero,Z);
            end

            if direction == 0
                steps = steps_ab(1,:);
            else
                steps = steps_ab(2,:);
            end

            for i = 0:length(steps)-2    % based on the direction, pick
the
                                         % section we will be
evaluating
                % Bisection starts here

                % Getting the first 2 limits for bisection
                top = steps(end-i-1);
                bot = steps(end-i);

                P_top = Evolve_P(P_temp,top,ab_index,direction).*Z;
                P_top = Evolve_P(P_top,ab_zero(2-
direction),ab_index,1-direction).*Z;
                y_top = sum(sum(P_top.*density_plane));

                P_bot = Evolve_P(P_temp,bot,ab_index,direction).*Z;
```

```
                P_bot = Evolve_P(P_bot,ab_zero(2-
direction),ab_index,1-direction).*Z;
                y_bot = sum(sum(P_bot.*density_plane));

                flag = 1;

                while flag == 1
                    mid = round((top+bot)/2);
                    if mid == top || mid == bot % best possible result
                        flag = 0;
                    else
                        P_mid =
Evolve_P(P_temp,mid,ab_index,direction).*Z;
                        P_mid = Evolve_P(P_mid,ab_zero(2-
direction),ab_index,1-direction).*Z;
                        y_mid = sum(sum(P_mid.*density_plane));

                        % Bisection conditions
                        if y_top > y_bot
                            if y_mid > yf
                                top = mid;
                            else
                                bot = mid;
                            end
                        else
                            if y_mid > yf
                                bot = mid;
                            else
                                top = mid;
                            end
                        end

                    end
                end

                P_top = Evolve_P(P_temp,top,ab_index,direction).*Z;
                P_top = Evolve_P(P_top,ab_zero(2-
direction),ab_index,1-direction).*Z;
                y_top = sum(sum(P_top.*density_plane));

                P_bot = Evolve_P(P_temp,bot,ab_index,direction).*Z;
                P_bot = Evolve_P(P_bot,ab_zero(2-
direction),ab_index,1-direction).*Z;
                y_bot = sum(sum(P_bot.*density_plane));

                % max_index - ab_index that correspond to the desired
input
                % P_max     - final Preisach plane state
                % yf2       - final outup, for calculated input
                if abs(y_bot-yf)<abs(y_top-yf)
                    max_index = bot;
                    P_max = P_bot;
                    yf2 = y_bot;
                else
                    max_index = top;
                    P_max = P_top;
                    yf2 = y_top;
                end

                results{(length(steps)-1)*direction + i + 1 + 4*reset,
1} = direction;
```

```
                results{(length(steps)-1)*direction + i + 1 + 4*reset,
2} = P_max;
                results{(length(steps)-1)*direction + i + 1 + 4*reset,
3} = yf2;

                error = (yf2-yf); % error for the calculated input
                results{(length(steps)-1)*direction + i + 1 + 4*reset,
4} = error;

                % Saves the calculated input
                if direction == 0
                    results{(length(steps)-1)*direction + i + 1 +
4*reset, 5} = alpha(max_index);
                else
                    results{(length(steps)-1)*direction + i + 1 +
4*reset, 5} = beta(max_index);
                end
                results{(length(steps)-1)*direction + i + 1 + 4*reset,
6} = reset;
            end
        end
    end
    results;

    % Check which is the best input, with the smallest absolute error
    E = [results{:,4}];
    error_index = find(min(abs(E))==abs(E));
    error_index = error_index(1);

    P_final = results{error_index,2};
    yf_calc = results{error_index,3};
    error = results{error_index,4};
    input = results{error_index,5};

    reset_flag = results{error_index,6};

    time = toc(tstart); % ends clock

end
```

C.4 - Generating the Error distributions

```
clear all
clc

% Defining Density Plane
==================================================
load('Marco_Fit_200.mat')
density_plane = Marco_density_plane{1};
beta = Marco_density_plane{2};
alpha = rot90(beta);
offset = Marco_density_plane{3};

% Random Desired Displacements
==========================================

y_max = -230;   % these valus were empiricly obtained, by observing
the
y_min = -800;   % outer butterfly loop

yf_desired = [];
yf_calc = [];
initial_P = {};
time = [];
error_dist = [];
u = [];

% Calculating optimum inputs
================================================

steps_check = 0;
steps_ab = [];
R = 0;
N = 64;     % Size of the square actuator grid,

for m = 1:N*N

    yf_desired(m) = (y_max-y_min)*rand()+y_min; % random desired
remnant
                                                % deformation

    Pi = Random_Preisach_Plane(alpha, beta);    % random initial
Preisach
                                                % Plane
%     initial_P{m} = Pi;  % only needed for debugging

    [u(m), Pf, yf_calc(m), error_dist(m), time(m),steps_check,
steps_ab, reset] = Find_max_input(density_plane, alpha, beta, Pi,
yf_desired(m), steps_check, steps_ab);

    R = R + reset;  % saves the ammount of times it was requested to
reset
                    % the material's memory to achieve the best result

%     final_P{m} = Pf;    % only needed for debugging

end

average_error = mean(error_dist)    % average error compared to
remnant
```

```
                                    % deformation bandwidth, should be
                                    % equal(close)to 0

standard_deviation = sqrt(sum((error_dist-
average_error).^2)/(length(error_dist)-1))
                                    % standar deviation for the error
                                    % distribution, result presented
in
                                    % nanometers

std_dev_perc = (standard_deviation/(y_max-y_min))*100
                                    % standard deviation for the error
                                    % distribution, result presented
in
                                    % percentage compared to remnant
                                    % deformation bandwidth
one_percent = abs(y_max-y_min)*0.01;
less_one_percent_error = sum(abs(error_dist)<one_percent)/(N*N)*100
                                    % percentage of the result which
have
                                    % less that 1% of error of the
                                    % bandwidth
two_percent = abs(y_max-y_min)*0.02;
less_two_percent_error = sum(abs(error_dist)<two_percent)/(N*N)*100
                                    % percentage of the result which
have
                                    % less that 2% of error of the
                                    % bandwidth

total_time = sum(time)              % total elapsed time

%% Plots
=========================================================================

% Error histogram (percental compared to bandwidth)
figure(1)
histogram((error_dist/(y_max-y_min))*100,
21,'Normalization','probability')
xlabel('Error (% of the bandwidth)')
title('Error distribution')
axis([-4*std_dev_perc 4*std_dev_perc 0 0.35])
legend(strcat('n = ', int2str(length(alpha))))
grid on
hold on
rectangle('position',[-1 0 2 1], 'FaceColor', [0 1 0 0.2])
rectangle('position',[-2 0 4 1], 'FaceColor', [1 1 0 0.2])
hold off
```

References

[1] - NASA/JPL-Caltech , "NASA Exoplanet Arquive", https://exoplanetarchive.ipac.caltech.edu/index.html (10/05/2019)

[2] – D. A. Fischer, A. W. Howard, G. P. Laughlin, B. Macintosh, S. Mahadevan, J. Sahlmann, J. C. Yee. "Exoplanet Detection Techniques" In: *Protostar and Planets VI*, Heidelberg, Germany, 2013

[3] - G. Chauvin, A. M. Lagrange, C. Dumas, B. Zuckerman, D. Mouillet, I. Song, J. L. Beuzit, P. Lowrance. "A giant planet candidate near a young brown dwarf" In: *Astronomy and Astrophysics* 425, L 29-32, 2004

[4] – M. Julien, M. Dimitri, M. David, K. Markus, G. Julien. "Adaptive optics in high-contrast imaging" In*: Cornell University Library*, 1701.00836v1, 2017

[5] – B. Lyot. "The study of the solar corona and prominences without eclipses" In: *Monthly Notices of the Royal Astronomical Society,* Vol. 99, 1939

[6] - D. Spergel, N. Gehrels et al., "WFIRST-AFTA Final Report", 2013

[7] - C. de Jonge, “Concept of a deformable mirror,” from Internal HDM group reports.

[8] - P. Curie, J. Curie, " Lois du degagement de l'electricite par pression, dans la tourmaline", in *Comptes Rendus* 92:186-188, 1880

[9] D. Skoog, F. Holler S. Crouch, " Principles of Instrumental Analysis", 6th edition, *Thoivision + Brooks/Cole*, 2007, page 9

[10] - J. Ewing , "The Strength of Materials", in *Cambridge University Press*, 1899

[11] - B. Drinčić et al, "Why are some hysteresis loops shaped like a butterfly?", in *Automatica*, 2011

[12] - L. Kuik, " Remnant Control of Piezoelectric Actuators in Hysteretic Deformable Mirrors", Bachelor Thesis at the University of Groningen, 2017

[13] - F. Preisach, "Über die magnetische nachwirkung", in *Zeitschrift für physik*, 94 (5-6), 277–302., 1935

[14] - W. van de Beek, " Operator-Based Modeling of Single Loop and Butterfly Hysteresis Phenomena in Piezoelectric Actuators", Master Thesis at the University of Groningen, 2018

[15] - W. Press et al., "Numerical Recipes in C: The Art of Scientific Computing", 2nd Edition, *Cambridge University Press*, Chapter 9, 1992

Image references

Figure 1: NASA/JPL-Caltech, "Exoplanet Plots" https://exoplanetarchive.ipac.caltech.edu/exoplanetplots/ (10/05/2019)

Figure 2(top): (Unknown Author) "WASP-14b", http://exoplanets.org/detail/WASP-14_b (10/05/2019)

Figure 2(bottom): N. Smolenski, "Planetary Transit", https://en.wikipedia.org/wiki/Methods_of_detecting_exoplanets#Transit_method (10/05/2019)

Figure 3: M. Kenworthy , "Direct Imaging of Exoplanets" https://home.strw.leidenuniv.nl/~kenworthy/direct_imaging.html (10/05/2019) (adapted)

Figure 4: Wikipedia user '2pem' , " Deformable mirror correction", https://en.wikipedia.org/wiki/Deformable_mirror (10/05/2019) (adapted)

Figure 5: J. Ewing, “Experimental researches in magnetism,” in *Philosophical Transactions of the Royal Society of London*, vol. 176, 1885, page 645.

Figures 7, 8, 9 and 10: C. de Jonge, “Concept of a deformable mirror,” from Internal HDM group reports.

Figures 11: B. Jayawardhana et al., "Modeling and analysis of butterfly loops via Preisach operators and its application in a piezoelectric material", in *57th IEEE Conference on Decision and Control*, pages 6894-6899, 2019

Figure 12 - W. van de Beek, " Operator-Based Modeling of Single Loop and Butterfly Hysteresis Phenomena in Piezoelectric Actuators", Master Thesis at the University of Groningen, 2018

Figure 14 - V. Hassani et al. Modeling hysteresis with inertial-dependent Prandtl-Ishlinskii model in wide-band frequency-operated piezoelectric actuator. *Smart Materials Research*, 2012 (adapted)

Figure 15 - W. S. Galinaitis, "Two methods for modeling scalar hysteresis and their use in controlling actuators with hysteresis" *Unpublished doctoral dissertation in Virginia Tech*, 1999 (adapted)

Figure 17 - J. Carstensen, " Interval Bisection Method ", https://www.tf.uni-kiel.de/matwis/amat/comp_math/kap_1/backbone/r_se19.html (15/5/2019)

[Blank Page]

[Blank Page]