

MODELING AND CONTROL OF MCKIBBEN ARTIFICIAL
MUSCLE – APPLICATION OF MODEL PREDICTIVE CONTROL,
GENETIC ALGORITHM AND METAHEURISTIC OPTIMIZATION

MICHELE IESSI

Master's Degree in Computer and Systems Engineering
Department of Information Engineering, Computer Science and Mathematics
University of L'Aquila

February 2019

Michele Iessi: *Modeling and Control of McKibben Artificial Muscle – Application of Model Predictive Control, Genetic Algorithm and Metaheuristic Optimization*, , © February 2019

SUPERVISORS:

Costanzo Manes

Kazuhisa Ito

ACKNOWLEDGEMENTS

Put your acknowledgements here.

CONTENTS

I	THESIS	1
1	INTRODUCTION	2
1.1	Background and Motivation	2
1.2	Modeling of McKibben Artificial Muscle	2
1.3	Aim of the Study	3
2	COMPOSITION OF THE THESIS	4
3	TAP WATER-DRIVEN MCKIBBEN ARTIFICIAL MUSCLE	6
4	MODELLING OF MCKIBBEN ARTIFICIAL MUSCLES	7
4.1	Introduction	7
4.2	Static Model	8
4.2.1	Introduction	8
4.2.2	Static Model of the Muscle	9
4.2.3	Modified Static Model	10
4.2.4	Static Model Validation	10
4.3	Muscle Model Based on System Identification	11
4.3.1	Identification Experiments	11
4.4	Hysteresis Modeling	14
4.5	Models of Hysteresis	14
4.6	The Bouc-Wen Model of Hysteresis	15
4.6.1	Classic Bouc-Wen Hysteresis Model	15
4.6.2	Generalized Bouc-Wen Hysteresis Model	15
4.7	Hysteresis Variable Approximation	16
4.7.1	Classic Bouc-Wen Model	16
4.7.2	Generalized Bouc-Wen Model	16
4.8	Identified Model Summation	17
4.8.1	Classic Bouc-Wen Model	17
4.8.2	Generalized Bouc-Wen Model	17
4.9	Evaluation of Proposed Models	17
4.9.1	Classic Bouc-Wen Model Parameters	18
4.9.2	Generalized Bouc-Wen Model Parameters	19
5	HYSTERESIS PARAMETER OPTIMIZATION	21
5.1	Introduction	21
5.2	Evolutionary Algorithms	21
5.2.1	Genetic Algorithm	22
5.2.1.1	Algorithm Overview	23
5.2.1.2	Algorithm Description	23
5.2.1.3	Initialization Step	23
5.2.1.4	Evaluation Step	24
5.2.1.5	Crossover Step	25
5.2.1.6	Mutation Step	26
5.3	Metaheuristic Algorithms	26
5.3.1	Firefly Algorithm	27
5.3.1.1	Algorithm Overview	27
5.3.1.2	Algorithm Description	28
5.3.1.3	Light Intensity and Attractiveness	28

5.3.1.4	Firefly Movement	29
5.3.2	Modified Firefly Algorithm	29
5.3.2.1	Attractiveness at Zero Distance	29
5.3.2.2	Light Absorption Coefficient	30
5.3.2.3	Randomization Parameter	30
5.3.3	Particle Swarm Optimization	31
5.3.3.1	Algorithm Overview	31
5.3.3.2	Algorithm Description	31
5.3.3.3	Velocity and Position Update	32
5.3.3.4	Inertia Weight	32
5.3.3.5	Acceleration Constants	32
5.4	Algorithm Comparison	32
5.4.1	System Specifications	33
5.4.2	Solution Comparison	33
5.4.2.1	Genetic Algorithm – Classic Bouc-Wen Model	33
5.4.2.2	Modified Firefly Algorithm – Classic Bouc-Wen Model	37
5.4.3	Time Comparison	38
6	CONTROLLER DESIGN	39
7	CONCLUSIONS AND FUTURE WORK	40
II	APPENDICES	41
A	SCRIPTS AND ALGORITHMS	42
A.1	Scripts	42
A.1.1	Stair Input Generation	42
A.2	Optimization Algorithms	42
A.2.1	Genetic Algorithm	42
A.2.2	Modified Firefly Algorithm	42
A.2.3	Particle Swarm Optimization	42
B	BOUC-WEN MODEL OF HYSTERESIS	43
	BIBLIOGRAPHY	44

LIST OF FIGURES

Figure 1	Geometric structure of a McKibben artificial muscle	8
Figure 2	Input waves for proportional valves	11
Figure 3	Pressure and displacement characteristic . . .	12
Figure 4	Pressure and displacement characteristic . . .	12
Figure 5	Transfer function fit	13
Figure 6	Comparison between experimental data and simulated output	18
Figure 7	Comparison between experimental data and simulated output with classic Bouc-Wen model . .	19
Figure 8	Comparison between experimental data and simulated output with generalized Bouc-Wen model	20
Figure 9	Genetic Algorithm – Classic BW Model (Random crossover)	34
Figure 10	Genetic Algorithm – Classic BW Model (Weighted crossover)	35
Figure 11	Genetic Algorithm – Classic BW Model (Proportional crossover)	36
Figure 12	Hysteresis loop result using genetic algorithm's parameters (proportional crossover approach)	37
Figure 13	Modified Firefly Algorithm – Classic BW Model	38

LIST OF TABLES

Table 1	Parameters for the Classic Bouc-Wen Model	18
Table 2	Parameters for the Generalized Bouc-Wen Model	19
Table 3	Parameters for the genetic algorithm	22
Table 4	Parameters for the genetic algorithm	28
Table 5	Parameters for the genetic algorithm	33
Table 6	Random crossover – Final values (Classic BW model)	34
Table 7	Weighted crossover – Final values (Classic BW model)	35
Table 8	Weighted crossover – Final values (Classic BW model)	36
Table 9	Parameters for the genetic algorithm	37

ACRONYMS

PAM	Pneumatic Artificial Muscle
PID	Proportional-Integrative-Derivative
MPC	Model Predictive Control
EA	Evolutionary Algorithm
GA	Genetic Algorithm
FA	Firefly Algorithm
MFA	Modified Firefly Algorithm
PSO	Particle Swarm Optimization

Part I

THESIS

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The current state of the art in hydraulic actuators consists almost entirely of oil driven valves, pistons and motors. However, this kind of actuator cannot be used in some particular applications, such as power assist systems and rehabilitation: they have significant heaviness and rigidity, because of their mechanical structure and motorization [1]. In this context, it is problematic to share a robot working space with humans around it, and so this kind of actuator cannot be used to actuate, for example, orthotics.

McKibben muscles were invented by Joseph L. McKibben to motorize pneumatic art orthotics. They in general consist of an inner rubber tube enclosed in a braided outer nylon sleeve. These muscles can be used as actuators of rehabilitation systems due to the following advantages:

- Light weight
- High power to weight ratio
- High flexibility
- Low cost
- Low environmental impact

However, there are drawbacks: it is well known that the muscle has poor control performance due to the existence of strong nonlinearities, such as hysteresis, cavitation and saturation characteristics. Furthermore, the wear of the materials (nylon sleeve and rubber tube) may cause a shorter lifetime with respect to other actuators.

1.2 MODELING OF MCKIBBEN ARTIFICIAL MUSCLE

As already mentioned, the control of McKibben muscles is not easy to achieve. A PID control solution may be developed, but it has flaws: the parameters of the controller will have to be tuned different types of muscles and for various loads. This is generally not an acceptable solution for this kind of application.

Thus, model-based control techniques are better suitable for this job. The plant model needs to be very precise for the control to be effective, and to do so identification techniques are used to get a first linear approximation of the model. Later, a hysteresis component is added to this linear model, to keep track of the nonlinearities added by the hysteretic behaviour of the muscle. Adding a hysteresis component to a linear model makes it more complex, so the choice of an appropriate hysteresis modeling technique is crucial to achieve good control performance.

This allows to get a model having good fit with respect to the real one. After this, it is possibly to try applying a model-based control approach, namely the **Model Predictive Control** (MPC). Further details about the development of controllers are in Chapter ??, and the theory behind MPC can be found in Appendix ??.

1.3 AIM OF THE STUDY

The aim of this dissertation is to get precise control of the displacement for a tap water-driven McKibben artificial muscle. To do so, a list of steps will be followed:

1. *Derivation of a simple linear model*

Using linear identification techniques, it is possible to map the input pressure to the output displacement, and get a precise yet simple linear model of the muscle. While in pneumatic artificial muscles it is required to take into account the temperature dynamics and the compressibility of air, using water as the muscle allows to disregard them. This grants a simpler model of the muscle.

2. *Introduction of hysteresis component*

The linear identification method grants a very simple yet linear model that does not take nonlinearities into account. The biggest source of nonlinearity for the McKibben muscle is hysteresis. A hysteresis component is added to the linear model, so that it will better follow the actual behaviour of the muscle. This step is essential to get good control performance.

3. *Hysteresis parameter estimation using optimization algorithms*

Evolutionary algorithms and metaheuristic optimization can be used in order to find suitable parameters for the hysteresis model. This procedure is useful to grant a good fit between real and simulated data, so that it is possible to make use of a precise plant model in a model-based control.

4. *Controller Design*

The application of Proportional-Integrative and Model Predictive Control techniques are proposed, using the muscle model obtained from linear system identification and modified with the introduction of the hysteresis component.

COMPOSITION OF THE THESIS

This dissertation is divided into chapters as follows.

Chapter 3 describes the McKibben artificial muscle, its general applications as well as the pros and drawbacks of using this kind of actuator.

Chapter 4 concerns the process of obtaining a simplified model from real data through model identification techniques. After this, a hysteresis component is introduced in the model, namely the Bouc-Wen model of hysteresis, to grant a better fit between simulated and real data. This step is essential so that this model can be used together with model-based control techniques so that it is possible to precisely control the muscle's position. Two different versions of the Bouc-Wen model have been studied and introduced, and their differences are highlighted in this Chapter.

Chapter 5 introduces the concept of *Evolutionary Algorithm* (EA) and *Metaheuristic Optimization* (MO). Three methods belonging to these categories of optimization techniques have been developed in order to find the best selection of parameters for the hysteresis model, and grant the best fit possible.

- **Evolutionary Algorithms** simulate the process of evolution that characterizes biological systems.

A *Genetic Algorithm* [2] (GA) approach has been studied and implemented to find a set of parameters that could provide a good fit between simulated and experimental values.

- **Metaheuristic Optimization** (MO) algorithms do not guarantee that a globally optimal solution may be found, but are designed to find, generate or select a set of parameters that may provide a sufficiently good solution to an optimization problem. Metaheuristics sample a set of solutions which is too large to be completely sampled. One of the main advantages of this method is that they may be used for a variety of problems with high performance.

Two methods belonging to this category have been studied and developed: the *Firefly Algorithm* [3], which mimics the behaviour of fireflies, and *Particle Swarm Optimization* [4], which iteratively tries to improve a candidate solution with regard to a given measure of quality.

The performance of each algorithm, based both on calculation time and fit result, is compared in Sections 5.4.3 and 5.4.2.

Chapter 6 covers the development and application of a PI control, as well as the tentative development of a model predictive control.

Chapter 7 lists the conclusions and gives mentions for future works. A series of appendices follows the chapters.

Appendix [A](#) contains all scripts and algorithms code used for this work, all thoroughly described.

Appendix [B](#) covers the Bouc-Wen model of hysteresis.

TAP WATER-DRIVEN MCKIBBEN ARTIFICIAL MUSCLE

MODELLING OF MCKIBBEN ARTIFICIAL MUSCLES

In this Chapter, the muscle models used as nominal models for model-based control are derived.

4.1 INTRODUCTION

McKibben muscles are actuators used mainly for medical purposes in rehabilitation and welfare. The reason is their high flexibility, light weight, low cost, human and environmental friendliness and ease of use. As mentioned in Chapter 1, the control of this kind of actuators is often problematic, due to their inherent high nonlinearity.

Many muscle models have been proposed, both static and dynamic. One of the most known static models is derived by Chou [5], which is based on the equilibrium between the input pressure and the release of energy. Although the main interest is to obtain the dynamic model to use in control, the static one is also reviewed and evaluated because they are used with feed-forward control applied to rehabilitation devices using the McKibben muscle.

Dynamic muscle models can be categorized in analysis oriented and control oriented. Analysis oriented models provide very high accuracy, but they are also very complex. For this reason they are not much suitable for control purposes.

Control oriented models provide lower accuracy than analysis oriented ones, but their lower complexity allows them to be used more efficiently for control.

Since tap water driven muscles are simpler than pneumatic ones, the idea is to use linear system identification and obtain a simple model of the muscle's dynamics. Being this only a linear model, it does not take into account the presence of strong nonlinearities, such as the friction between the braids and the hysteretic behaviour of the muscle. However, the introduction of a hysteresis model can lead to achieving higher precision with the model derived from linear system identification.

Through history, several hysteresis models have been developed. Notable examples are the Maxwell-slip model [6], the Jiles-Atherton model [7] and the Preisach model [8]. Common interest points of these models are friction of the braided sleeve, and the hysteresis caused by it, which both add nonlinearities to the model.

These hysteresis models are precise, yet complex in structure. To overcome this problem, the Bouc-Wen [9] [10] hysteretic model is combined with the identified muscle model. Including the hysteretic model raises the number of parameters that have to be identified for the system, but this is easily achieved, at first, by trial and error. Later, an adaptive algorithm is developed to get the best parameters for the

muscle, and thus the updated model can be use as a nominal model for model-based control techniques.

The accuracy of the proposed model is compared to experimental results by analysing the pressure–displacement characteristics and the hysteresis characteristics. Moreover, the effects of changing the load of the muscle are studied, because they may have a great impact upon control performances.

4.2 STATIC MODEL

4.2.1 Introduction

The relationship between the axial contraction force and the pressure difference amid supply and atmospheric pressure has been reported in different papers [11] [12].

The association is based on the equilibrium between the input work in the McKibben muscle (i. e. when the fluid is supplied to the inner rubber tube) and the output work (i. e. when the actuator shortens or elongates because of the volumetric change associated with the pressure difference).

Figure 1 shows the geometric structure, which follows the geometric relationships in Equation 1.

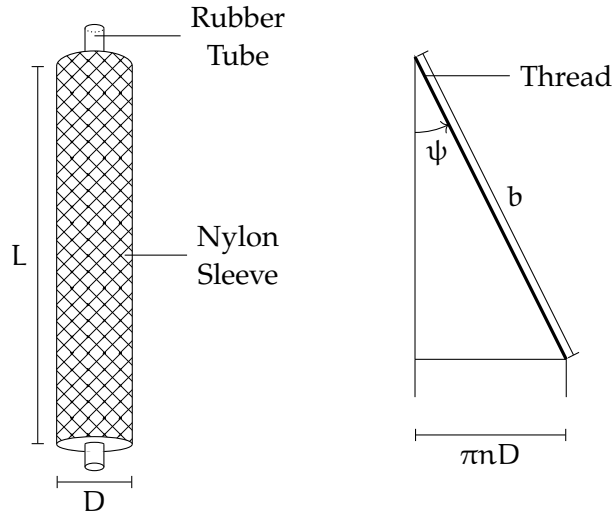


Figure 1: Geometric structure of a McKibben artificial muscle

$$L = b \cos(\psi), \quad D = \frac{b \sin(\psi)}{n\pi} \quad (1)$$

where L is the length of the muscle, D is the outer diameter of the muscle, b is the thread length, n is the number of turns of a thread, and ψ is the angle formed by the sleeve's threads and its vertical axis.

4.2.2 Static Model of the Muscle

The input work W_{in} is applied to the muscle when the fluid (liquid or air) pushes the internal surface of the rubber tube. This can be expressed as the product of the supply pressure and the change in volume.

$$dW_{in} = (P - P_0) dV = P' dV \quad (2)$$

Where P is the supply pressure, P_0 is the atmospheric pressure, P' is the pressure difference and dV is the volumetric change.

Equation 3 shows the volume of the muscle, with the assumption that it has cylindrical shape.

$$V = \frac{1}{4} \pi L D^2 \quad (3)$$

Then, from Equation 1,

$$V = \frac{1}{4} \pi L D^2 = \frac{b^3}{4\pi n^2} \sin(\psi)^2 \cos(\psi) \quad (4)$$

When the actuator contracts, the output work of the system W_{out} is given by Equation 5.

$$dW_{out} = F \times (-dL) \quad (5)$$

Where F is the axial tension, and dL is the axial displacement change.

From Equation 1 we can express $dL/d\psi$ as follows.

$$\frac{dL}{d\psi} = -b \sin(\psi) \quad (6)$$

With the assumption that the input work and the output work should be equal if the system is lossless and without energy storage, as for Equations 2, 4, 5 and 6, then the relation shown in Equation 7 can be obtained.

$$F = -P' \frac{dV/d\psi}{dL/d\psi} = \frac{\pi P'}{4} \left(\frac{b}{\pi n} \right)^2 (3 \cos^2(\psi) - 1) \quad (7)$$

From this, it can be observed that the tension F is linearly proportional to the pressure difference P' , and it is a monotonic function of the angle ψ ($0^\circ < \psi < 90^\circ$).

4.2.3 Modified Static Model

The static model of a McKibben muscle can be expressed as Equation 7. However, that model does not take into account the thickness t_k of both the nylon sleeve and the internal rubber tube. If this is considered, then the volume calculated in Equation 3 becomes the following.

$$\begin{aligned} V &= \frac{1}{4}\pi L (D - 2t_k)^2 \\ &= \frac{b^3}{4\pi n^2} \sin(\psi) (3 \cos^2(\psi) - 1) - b \cos(\psi) t_k \left(\frac{b}{n} \sin(\psi) - \pi t_k \right) \end{aligned} \quad (8)$$

From Equation 8 then it is possible to obtain the equivalent of Equation 7 with thickness included. The result is shown in Equation 9.

$$F = \frac{b^2 P'}{4\pi n^2} (3 \cos^2(\psi)) + \pi t_k P' \left[\frac{b}{\pi n} \left(2 \sin(\psi) - \frac{1}{\sin(\psi)} \right) - t_k \right] \quad (9)$$

4.2.4 Static Model Validation

It is possible to validate the static model obtained. Stepwise voltage is applied to the input and output proportional valves, to study the behaviour of the muscle under quasi static conditions.

(Aggiungere validazione del modello con enfasi sull'errore derivante attriti e approssimazioni)

4.3 MUSCLE MODEL BASED ON SYSTEM IDENTIFICATION

If all sources of non linearity could be taken into account, then theoretical models may agree with experimental results. However, by taking into consideration these elements, the models would become very complex and they would be difficult to control. Moreover, the models would lack in versatility, meaning that these parameters would have to be changed for different muscles and weights.

The model of the muscle is thus obtained through linear system identification techniques and tools such as MATLAB and Simulink. In this section the derivation of the muscle model based on system identification is described, as well as the improvement of the identified model by using the Bouc-Wen model of hysteresis.

4.3.1 Identification Experiments

For the identification step, a stepwise signal has been generated and given to the input and output proportional valves. The two signals are symmetric one another, due to the nature of the valves. They range from 0 to 10 Volts, and are shown in figure 2. The algorithm for generating the signals is in Appendix A (Algorithm A.1.1).

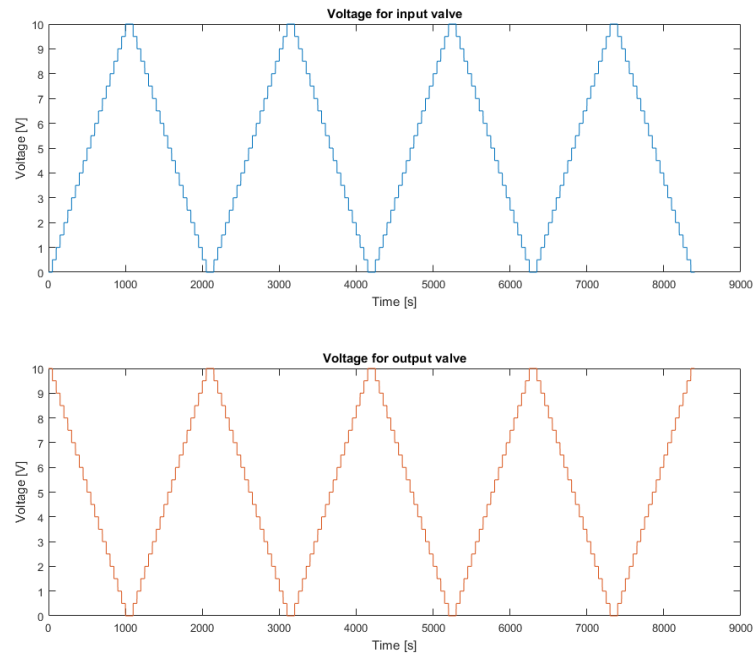


Figure 2: Input waves for proportional valves

Figure 3 shows the pressure and displacement characteristics of the muscle with the input just introduced.

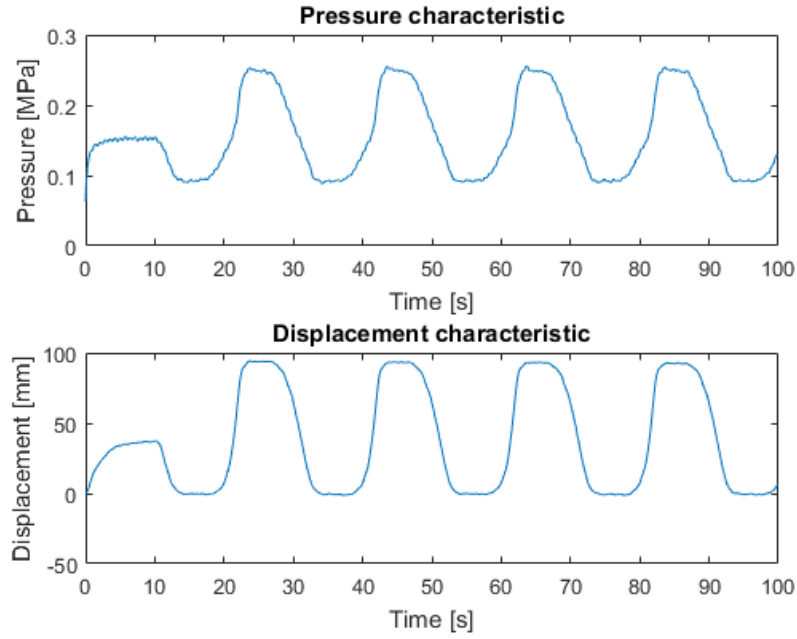


Figure 3: Pressure and displacement characteristic

By plotting this data with pressure on the x-axis and displacement on the y-axis, it is possible to notice the hysteretic behaviour of the muscle, as shown in Figure 4.

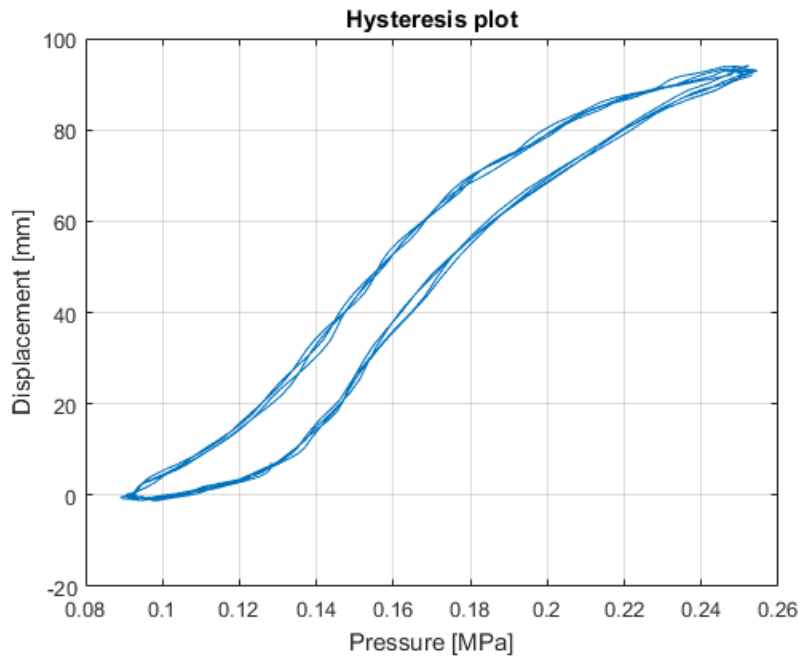


Figure 4: Pressure and displacement characteristic - Hysteresis plot

Using the MATLAB System Identification Toolbox [13], we can get a discrete transfer function from input and output values. The description on how to use the toolbox is provided in Appendix ??.

The input and output data are given to the toolbox. After that, input and output data are treated in the following way:

- their mean is put to 0
- the data is split in half and used in the following way:
 - the first half is used for the *estimation process*, by which the toolbox attempts to find a discrete function that correctly approximates the input-output characteristic of the system.
 - the second half is used for the *validation process*, by means of which the toolbox computes the fit of the identified transfer function.

The toolbox permits to select the number of zeros and poles that the transfer function must present. A number of 1 poles and 1 zeros has been chosen for this step.

The identification step then provides the discrete transfer function in Equation 10, where $L(z)$ and $P(z)$ represent respectively the z transform of displacement and pressure.

$$G(z) = \frac{L(z)}{P(z)} = \frac{24.9366 z^{-1}}{1 - 0.9618 z^{-1}} \quad (10)$$

The fit of this transfer function with respect to experimental data is shown in Figure 5.

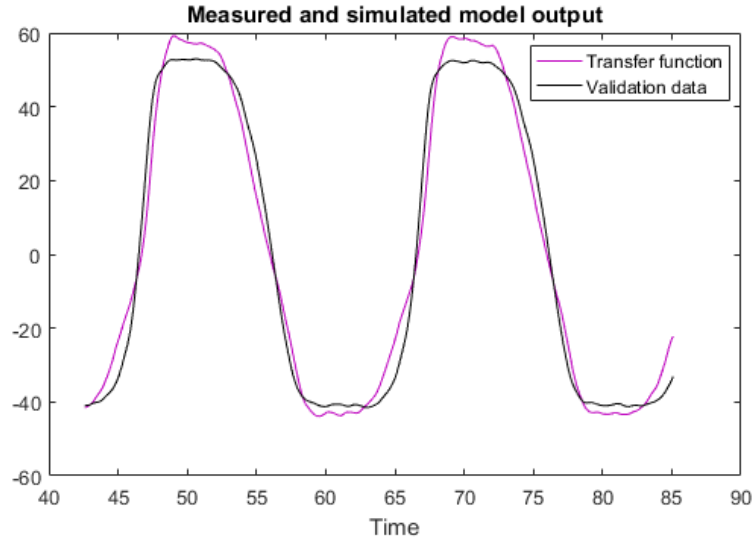


Figure 5: Transfer function fit

It is possible to raise the number of poles and zeros of the transfer function in order to improve its fit with respect to experimental data. The trade-off is that the resulting transfer function will be more complex.

The transfer function in Equation 10 can be written rewritten as follows

$$L(z) - 0.9618L(z)z^{-1} = 24.9366P(z)z^{-1} \quad (11)$$

This implies

$$\begin{aligned}
l(k) &= 0.9618l(k-1) + 24.9366p(k-1) \\
&= al(k-1) + bp(k-1) \\
a &= 0.9618, \quad b = 24.9366
\end{aligned} \tag{12}$$

So the displacement at step k depends on the displacement and pressure at the previous step.

4.4 HYSTERESIS MODELING

Hysteresis represents a property of systems that show the dependence of their state from their history. It is a nonlinear phenomenon exhibited by systems stemming from various science and engineering areas: under a low-frequency periodic excitation, the relationship between the system's input and output is not the same for loading and unloading. This means that the same input applied at different times can yield different output with respect to the history of the system.

The etymology of the term "hysteresis" is derived from the ancient Greek, and it means "delay" or "lagging behind".

A fundamental theory that allows a general framework for modeling hysteresis has not been developed yet.

For specific problems, models derived from a general understanding of physics are normally used. This kind of models are, however, hard to derive and difficult to use in practical application due to their complexity. For this reason, alternative models of these complex systems have been proposed. Phenomenological (or semi-physical) models are simpler models that, although not giving the best description of the behaviour of a system, combine some physical understanding of the system along with some kind of black-box modeling, keeping relevant input-output information that is useful for characterization, design and control purposes.

Aggiungere descrizione dettagliata isteresi e introdurre cosa si leggerà in seguito

4.5 MODELS OF HYSTERESIS

There is a large variety of hysteresis models available in literature, such as:

- the Preisach model [14], used to model electromagnetic hysteresis
- the Maxwell-Slip model [15], used for friction simulation and compensation,
- the Bouc-Wen model, extensively used in the areas of smart structures and civil engineering [10]

The latter consists of a first-order nonlinear differential equation that relates the input displacement to the output restoring force in a rate-independent hysteretic way [16].

4.6 THE BOUC-WEN MODEL OF HYSTERESIS

Proposed by Bouc [9] in 1971 and later modified by Wen [10] in 1976, the Bouc-Wen model has been vastly used to mathematically describe components and devices that present hysteretic behaviours, particularly within the areas of civil and mechanical engineering.

The differential equation consists of several parameters. By choosing them appropriately, it is possible to accommodate the response of the model to the real hysteresis loop [17]. The derivation of the model is available in Appendix B.

Equation 12 can be modified by adding a virtual hysteresis component. This should allow the hysteresis loop to adapt to the experimental values by changing the model's parameters.

$$l(k) = a [\alpha l(k-1) + (1-\alpha)w(k-1)] + bp(k-1) \quad (13)$$

$$\alpha \in [0, 1], \quad a = 0.9618, \quad b = 24.9366$$

Here w is a virtual hysteresis variable introduced by the use of Bouc-Wen model. It is possible to implement two different versions of the hysteresis model:

- the one proposed by Bouc and modified by Wen, that will be henceforth called *classic* version
- a more recent and flexible version proposed by Song and Der Kiureghian, that will be called *generalized* version

4.6.1 Classic Bouc-Wen Hysteresis Model

The classic Bouc-Wen model's virtual hysteresis variable w is described by Equation 14.

$$\dot{w} = A\dot{l} - \beta|\dot{l}||w|^{n-1} - \gamma\dot{l}|w|^n \quad (14)$$

A , β , γ and n are Bouc-Wen model's parameters. They control the shape and the slope of the hysteretic loop. Further details are described in Appendix B.

4.6.2 Generalized Bouc-Wen Hysteresis Model

Song and Der Kiureghian [18] proposed a generalized Bouc-Wen model for highly asymmetric hysteresis that in certain cases shown higher flexibility and resiliency than the classic Bouc-Wen model.

The generalized Bouc-Wen model's virtual hysteresis variable w is described by Equations 15 and 16.

$$\dot{w} = \dot{l} [A - |w|\Psi(l, \dot{l}, w)] \quad (15)$$

$$\begin{aligned} \Psi(l, \dot{l}, w) = & \beta_1 \operatorname{sgn}(\dot{l}w) + \beta_2 \operatorname{sgn}(\dot{l}) + \beta_3 \operatorname{sgn}(lw) \\ & + \beta_4 \operatorname{sgn}(\dot{l}) + \beta_5 \operatorname{sgn}(w) + \beta_6 \operatorname{sgn}(l) \end{aligned} \quad (16)$$

where β_1, \dots, β_6 are fixed parameters.

4.7 HYSTERESIS VARIABLE APPROXIMATION

Equations 14 and 15 should be approximated to a discrete time equation, as Equation 13 is in discrete time. To do so, it is possible to use forward difference approximation method, as shown in Equations 17 and 18. T_s represents the sampling time for the approximation.

$$\frac{dx}{dt} \approx \frac{x(t + T_s) - x(t)}{T_s} = \frac{x(k + 1) - x(k)}{T_s} \quad (17)$$

$$\frac{d^2x}{dt^2} \approx \frac{x(t + 2T_s) - 2x(t + T_s) + x(t)}{T_s^2} = \frac{x(k + 2) - 2x(k + 1) + x(k)}{T_s^2} \quad (18)$$

4.7.1 Classic Bouc-Wen Model

In the classic version of the Bouc-Wen model, the virtual hysteresis variable w can be discretized as follows.

$$\begin{aligned} \frac{w(k + 1) - w(k)}{T_s} = & A \frac{l(k + 1) - l(k)}{T_s} - \beta \left| \frac{l(k + 1) - l(k)}{T_s} \right| |w(k)|^{n-1} \\ & - \gamma \frac{l(k + 1) - l(k)}{T_s} |w(k)|^n \end{aligned} \quad (19)$$

Solving for $w(k)$ gives the result in Equation 20.

$$\begin{aligned} w(k) = & A [l(k) - l(k - 1)] - \beta |l(k) - l(k - 1)| |w(k - 1)|^{n-1} \\ & - \gamma [l(k) - l(k - 1)] |w(k - 1)|^n + w(k - 1) \end{aligned} \quad (20)$$

4.7.2 Generalized Bouc-Wen Model

In the generalized version of the Bouc-Wen model, the virtual hysteresis variable w can be discretized as follows.

$$\frac{w(k + 1) - w(k)}{T_s} = \frac{l(k + 1) - l(k)}{T_s} [A - |w(k)| \Psi(l, w, k)] \quad (21)$$

$$\begin{aligned} \Psi(l, w, k) = & \beta_1 \operatorname{sgn} \left(\left[\frac{l(k + 1) - l(k)}{T_s} \right] w(k) \right) \\ & + \beta_2 \operatorname{sgn} \left(\left[\frac{l(k + 1) - l(k)}{T_s} \right] l(k) \right) \\ & + \beta_3 \operatorname{sgn} (l(k)w(k)) + \beta_4 \operatorname{sgn} \left(\frac{l(k + 1) - l(k)}{T_s} \right) \\ & + \beta_5 \operatorname{sgn} (w(k)) + \beta_6 \operatorname{sgn} (l(k)) \end{aligned} \quad (22)$$

Solving Equation 21 for $w(k)$ gives the result in Equation 23.

$$\begin{aligned}
 w(k) &= [l(k) - l(k-1)] [A - |w(k)|\Psi(l, w, k)] + w(k-1) \\
 \Psi(l, w, k) &= \beta_1 \operatorname{sgn}([l(k) - l(k-1)] w(k)) \\
 &\quad + \beta_2 \operatorname{sgn}([l(k) - l(k-1)] l(k)) \\
 &\quad + \beta_3 \operatorname{sgn}(l(k)w(k)) + \beta_4 \operatorname{sgn}(l(k) - l(k-1)) \\
 &\quad + \beta_5 \operatorname{sgn}(w(k)) + \beta_6 \operatorname{sgn}(l(k))
 \end{aligned} \tag{23}$$

4.8 IDENTIFIED MODEL SUMMATION

The previous sections can be summarized with Equations 24 and 25.

4.8.1 Classic Bouc-Wen Model

$$\begin{cases}
 l(k) = a [\alpha l(k-1) + (1 - \alpha)w(k-1)] + bp(k-1) \\
 w(k) = A [l(k) - l(k-1)] - \beta |l(k) - l(k-1)| |w(k-1)|^{n-1} \\
 \quad - \gamma [l(k) - l(k-1)] |w(k-1)|^n + w(k-1) \\
 \alpha \in [0, 1], \quad a = 0.9618, \quad b = 24.9366
 \end{cases} \tag{24}$$

4.8.2 Generalized Bouc-Wen Model

$$\begin{cases}
 l(k) = a [\alpha l(k-1) + (1 - \alpha)w(k-1)] + bp(k-1) \\
 w(k) = [l(k) - l(k-1)] [A - |w(k)|\Psi(l, w, k)] + w(k-1) \\
 \Psi(l, w, k) = \beta_1 \operatorname{sgn}([l(k) - l(k-1)] w(k)) \\
 \quad + \beta_2 \operatorname{sgn}([l(k) - l(k-1)] l(k)) \\
 \quad + \beta_3 \operatorname{sgn}(l(k)w(k)) + \beta_4 \operatorname{sgn}(l(k) - l(k-1)) \\
 \quad + \beta_5 \operatorname{sgn}(w(k)) + \beta_6 \operatorname{sgn}(l(k)) \\
 \alpha \in [0, 1], \quad a = 0.9618, \quad b = 24.9366
 \end{cases} \tag{25}$$

4.9 EVALUATION OF PROPOSED MODELS

The models previously proposed are tested and evaluated to estimate the error with respect to experimental data.

Without taking the Bouc-Wen model into consideration, given experimental input values p , using equation 10 the simulated output \hat{l} with respect the experimental values l is shown in Figure 6.

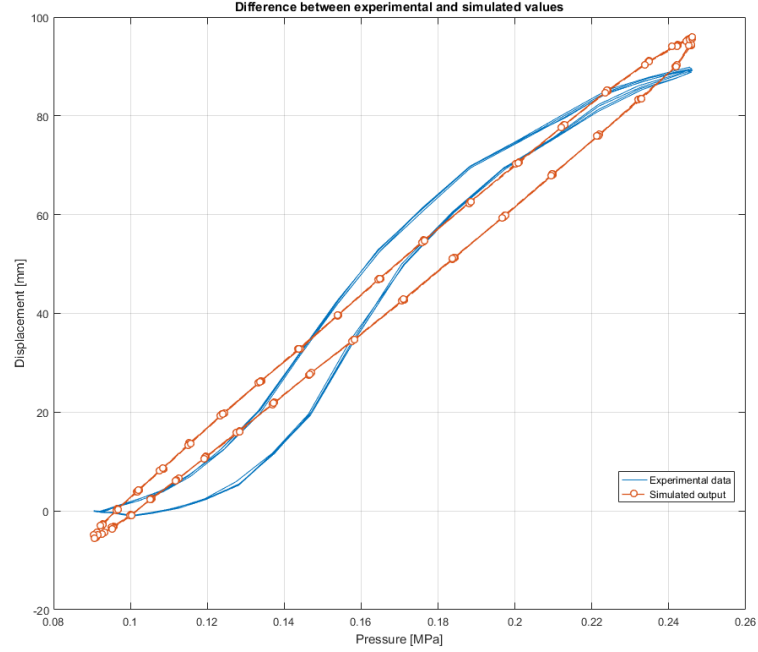


Figure 6: Comparison between experimental data and simulated output

If the Bouc-Wen model is taken into consideration, then the parameters must be fixed appropriately. To get a fairly good fit it is possible to tune these parameters by trial and error. Further methods for finding the best sets of parameters for the hysteresis model are discussed in Chapter 5.

4.9.1 Classic Bouc-Wen Model Parameters

The parameters to be chosen for the classic version of the Bouc-Wen model are α , A , β , γ and n . For this work, n will be fixed at 1 for both the classic and generalized version of the hysteresis model. Details about how each parameters alters the shape of the hysteretic loop are in Appendix B.

The parameters in Table 1 have been fixed by trial and error. All parameters are adimensional.

Parameter	Value
α	0.5
A	0.9
β	0.002
γ	-0.001
n	1

Table 1: Parameters for the Classic Bouc-Wen Model

Given experimental input p , the simulated output of the system taking into consideration the classic Bouc-Wen model of hysteresis is shown in Figure 7.

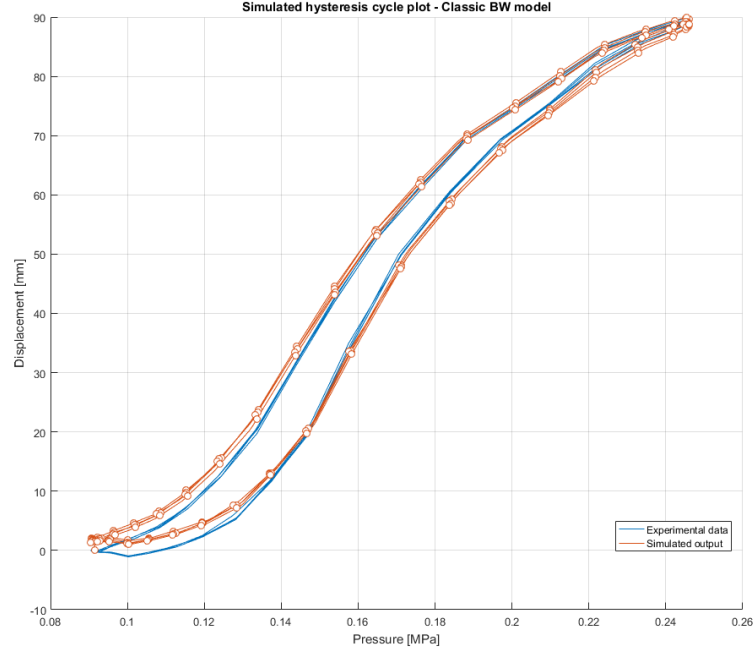


Figure 7: Comparison between experimental data and simulated output with classic Bouc-Wen model

4.9.2 Generalized Bouc-Wen Model Parameters

The parameters to be chosen for the generalized version of the Bouc-Wen model are α , A , n , β_1 , \dots , β_6 . As previously stated, the value of n is fixed to 1.

As for the classic version of the hysteresis model, the parameters in Table 2 have been selected by trial and error. All parameters are adimensional.

Given experimental input p , the simulated output of the system after taking into consideration the generalized Bouc-Wen model of hysteresis is shown in Figure 8.

Parameter	Value
α	0.9
A	1
n	1
β_1	0.01
β_2	0.005
β_3	0.1
β_4	-0.01
β_5	-0.1
β_6	0.001

Table 2: Parameters for the Generalized Bouc-Wen Model

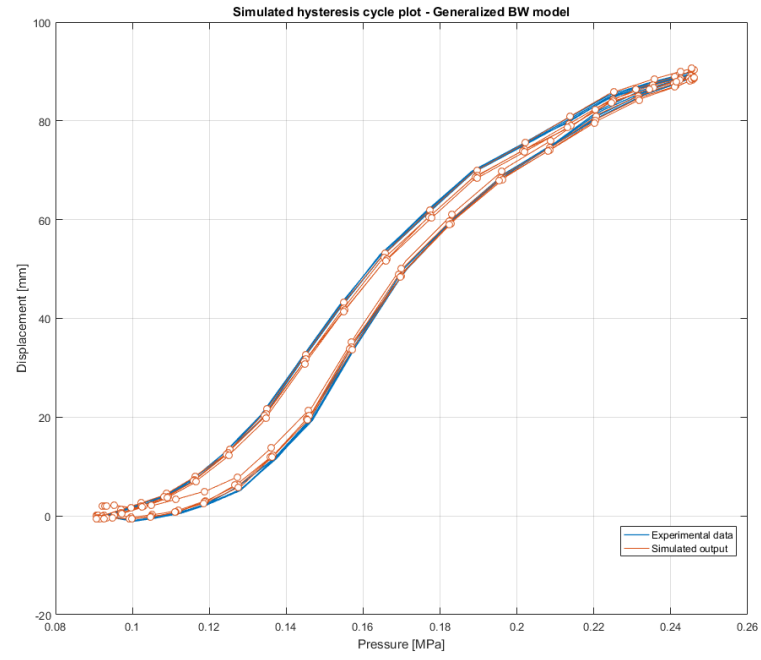


Figure 8: Comparison between experimental data and simulated output with generalized Bouc-Wen model

HYSTERESIS PARAMETER OPTIMIZATION

In this chapter, the methods to obtain the model's parameters are explained. As shown in Section 4.9, although parameters cherry-picked by trial and error proved successful to fit the approximated model onto experimental data, a higher grade of precision is desirable for applications demanding high accuracy.

Section 5.2 introduces the concept of *evolutionary algorithm*, with particular regard towards a simple genetic algorithm approach with the purpose of optimization.

After that, Section 5.3 introduces methods of *metaheuristic algorithm*, with the goal of transforming the search of parameters into an optimization problem.

Section 5.4 contains results and comparison for all algorithms previously described.

5.1 INTRODUCTION

After having described how to get a discrete model of the McKibben artificial muscle, the Bouc-Wen model of hysteresis has been introduced. This model proves useful to describe a hysteretic behaviour in a simple manner, and can provide great advantages in terms of performance when it comes to control.

However, there are parameters that have to be tuned in order for the model to operate correctly. The tuning may be done by trial and error, or the whole process may be automated.

The parameter refinement process makes use of an algorithm to find the parameters of the Bouc-Wen model that give a best approximation of the real experimental data. Such algorithms include, but are not limited to, Recursive Least Squares (RLS), evolutionary algorithms and multi-variable optimization algorithms.

For this work, different approaches based on evolutionary algorithms and multi-variable optimization methods have been tested.

5.2 EVOLUTIONARY ALGORITHMS

An evolutionary algorithm is a population-based optimization algorithm that mimics the behaviour of biological evolution, simulating the steps of reproduction, mutation, recombination and selection.

Each individual is tested in an optimization problem, and the best characteristic of the population are passed through the generations. The evaluation of an individual is done by means of a fitness function. A fitness function is an operator that assigns a fitness value to each individual. Depending on the nature of the problem, individuals with lower (or higher) fitness values are judged "better" from the algorithm

Parameter	Description
N	The population's size, i.e. the number of individuals (candidate solutions) that evolve to find a good approximation.
G	The maximum generation. After reaching that stage of evolution, the algorithm stops.
d	The chromosome's dimension, i.e. the number of variables to be optimized by the algorithm.
L	A d-dimensional vector containing the lower bounds for each variable.
U	A d-dimensional vector containing the upper bounds for each variable.
P_{cross}	The crossover probability. It controls the chance of inheriting a chromosome from a specific parent with respect to the other.
P_{mut}	The mutation probability.
m_m	The mutation magnitude, i.e. how much a given chromosome changes in the event of a mutation.

Table 3: Parameters for the genetic algorithm

and have a higher probability of being chosen in the mating step to generate new, better individuals. Therefore, their characteristics have a higher chance to be passed onto next generations in order to iteratively refine the parameters and arrive at a acceptable solution.

After a certain amount of generations, or after one individual reaches a fixed fit value, the algorithm stops and the best individual is chosen as the solution to the optimization problem.

5.2.1 Genetic Algorithm

A genetic algorithm (GA) [2] is a kind of evolutionary algorithm inspired by the concept of natural selection that permeates a biological evolutionary process.

Genetic algorithms are a class of stochastic global search algorithm operating on a set called *population* of current approximations, called *individuals*. Individuals are encoded as a set (*chromosome*) of parameters (*genes*).

Once a population has been created, each individual performance is assessed according to the objective function characterising the problem that has to be solved.

Algorithms belonging to this class are divided into the following steps: *initialization*, *evaluation*, *crossover* and *mutation*. The general outline of the procedure is shown in Algorithm 1. The description and peculiarities of each step are discussed in Section 5.2.1.2. The genetic algorithm also makes use of several parameters, which are summarized in Table 3.

5.2.1.1 Algorithm Overview

Algorithm 1 Genetic algorithm approach**procedure** GENETIC ALGORITHM**Input:** $N, G, d, L, U, p_{\text{cross}}, p_{\text{mut}}, m_m$ **Output:** A d -dimensional vector, i.e. the candidate solution (individual) with best fit value**Initialization:** generate N d -dimensional vectors $n_i, i \in [1, N]$ current_gen $\leftarrow 1$ **while** current_gen $< G$ **do****Evaluation:** compute the fit value $\text{fit}(i)$ for each vectorCompute the normalized fit value $\text{fit}_n(i)$ for each vector $M_p \leftarrow$ new empty Set**while** $|M_p| < N$ **do****Crossover:** extract 2 individuals

Generate 2 new individuals (offspring)

Add the offspring to M_p **end while****Mutation:****for each** $m \in M_p$ **do****for each** gene $m_i, i = 1, \dots, d$ **do**Generate a random value r such that $0 \leq r \leq 1$ **if** $r \leq p_{\text{mut}}$ **then**Mutate the i -th gene with percentage $\pm m_m \%$ **end while**

Return the individual with best fit value

5.2.1.2 Algorithm Description

5.2.1.3 Initialization Step

For a genetic algorithm, the initialization step consists in the creation of the individuals, and by extension that of the population.

One of the most common initialization methods is random initialization: each gene of any individual is randomly chosen between a defined interval. This gives a higher chance of finding good results from the first generations, and refine the solution starting from them.

Another initialization method may be to create a population of equal individuals, and mainly rely on mutation for the first generations in order to find better approximations. This method, while it may better reflect the natural behaviour of a biological evolutionary process, takes also a much larger number of generations to return appreciable results.

For this reason, the GA developed for this work makes use of random initialization. N d -dimensional vectors (individuals) p are created, following the constraints in Equation 26.

$$\begin{cases} p_i = [p_{i1}, \dots, p_{id}] \\ L_j \leq p_{ij} \leq U_j, \quad i = 1, \dots, N, \quad j = 1, \dots, d \end{cases} \quad (26)$$

5.2.1.4 Evaluation Step

The evaluation step concerns assessing a fitness value to each individual in order to rank them according to their performance in solving the optimization problem. The fitness value is assigned to each individual by means of a *fitness function*.

Experimental input pressure and output displacement, respectively described in Equations 27 and 28, are taken from data acquired by the identification experiments described in Section 4.3.1.

$$u = [u_1, u_2, \dots, u_k] \quad (27)$$

$$l = [l_1, l_2, \dots, l_k] \quad (28)$$

with k representing the number of points for which the experimental data has been sampled.

Each individual is used to perform a specific simulation of the model with its genes set as Bouc-Wen hysteresis parameters, using u as input. The simulated output displacement \hat{l} is then compared to the experimental displacement l .

The fitness function chosen for this step is described in Equation 29. This function returns the fitness value for the i -th individual of the population.

$$\text{fit}(i) = \frac{1}{\sum_{i=1}^k (l_i - \hat{l}_i)^2} \quad (29)$$

Judging from the nature of the fitness function, it is clear that individuals with a fitness value close to 0 are better than others, as it means that their simulated output is close to experimental data.

After that, the fit value of each individual is normalized in order to be used for the crossover step. To do so, each fit value is divided by the sum of all fit values, as shown in Equation 30.

$$\text{fit}_n(i) = \frac{\text{fit}(i)}{\sum_{i=1}^N \text{fit}(i)} \quad (30)$$

It is important to notice that given the way that fit_n is constructed, the relation of Equation 31 arises.

$$\sum_{i=1}^N \text{fit}_n(i) = 1 \quad (31)$$

5.2.1.5 Crossover Step

The crossover step serves the purpose of creating two new individuals (*offspring*) starting from two *parents*. The characteristics of the offspring will thus largely depend on their parents' ones.

Three different crossover policies have been developed for this purpose:

- **Random crossover**

The random method assigns the genes from parents to offspring in a random fashion.

Given a number x with $1 \leq x \leq d$, the offspring inherits the first x genes from the first parent and the remaining ones from the second parent. This process is repeated for the second offspring.

- **Weighted crossover**

The weighted method makes so that the genes of the parent with higher fit have a higher probability of being chosen for the offspring.

Given the fit values of the parents $\text{fit}(p_1)$ and $\text{fit}(p_2)$, with $\text{fit}(p_1) \geq \text{fit}(p_2)$, then the *inheritance probability* In_p is defined as follows:

$$\text{In}_p = \frac{\text{fit}(p_1)}{\text{fit}(p_1) + \text{fit}(p_2)}$$

For each offspring and gene, the probability of inheriting the given gene from the first parent will be In_p . Otherwise, the gene will be inherited from the second parent.

- **Proportional crossover**

The proportional method creates the offspring's genes from the parents' ones, proportionally inheriting them based on their fit values.

Given the fit values of the parents $\text{fit}(p_1)$ and $\text{fit}(p_2)$, with $\text{fit}(p_1) \geq \text{fit}(p_2)$, then the *proportion variables* pv_1 and pv_2 are defined as follows:

$$\text{pv}_1 = \frac{\text{fit}(p_1)}{\text{fit}(p_1) + \text{fit}(p_2)}, \quad \text{pv}_2 = 1 - \text{pv}_1$$

Given $\text{Gp}_1(x)$, $\text{Gp}_2(x)$, $\text{Go}(x)$, $1 \leq x \leq d$ respectively for the first parent's, the second parent's and the offspring's x -th gene, then the following relation arises:

$$\text{Go}(x) = \text{Gp}_1(x) \cdot \text{pv}_1 + \text{Gp}_2(x) \cdot \text{pv}_2$$

This means that the offspring will be more similar to the parent with higher fit, while still carrying on some of the less performing parent's characteristics.

5.2.1.6 Mutation Step

The mutation step's purpose is to randomly mutate some genes to reflect the random behaviour of biological evolution through time.

For each gene $Go(x)$, $1 \leq x \leq d$ of each offspring generated, a random value r between 0 and 1 is generated. If this value is less than the *mutation probability* P_{mut} , then the gene is modified by a percentage of its value.

The final value $Go(x)'$ of the x -th gene will be:

$$Go(x)' = Go(x) \cdot (1 \pm m_m)$$

Here the *mutation magnitude* is added or subtracted with probability 0.5, by generating a new random value and verifying whether it surpasses this threshold.

5.3 METAHEURISTIC ALGORITHMS

A *metaheuristic* is a procedure designed to find a heuristic that may provide a sufficiently good solution to an optimization problem.

Metaheuristic algorithms do not guarantee that a globally optimal solution can be found on some class of problems, but they often provide a solution with a good fit value by exploring the space of possible solutions.

For this work, two metaheuristic algorithms have been developed and tested with the purpose of finding good parameters for fitting the Bouc-Wen hysteresis parameters of the identified transfer function over real experimental data.

The first is the *Firefly Algorithm*, described in Subsection 5.3.1. This algorithm mimics the movement of a swarm of fireflies inside the solution space, simulating an *attractiveness component*, governed by their *luminescence*, that pushes the swarm towards local and global optimal solutions. Two versions of this algorithm have been developed in order to grant a higher versatility for finding the parameters, depending on the required application. The modified version is described thoroughly in Section 5.3.2.

The second algorithm is the *Particle Swarm Optimization*, described in Section 5.3.3. It is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions (*particles*) moving around in the search-space according to simple laws governed by their position and velocity. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space. This value is updated whenever a better position is found any particle. This is expected to move the swarm toward the best solutions. The algorithm has been conceived as a mathematical representation of the movement of organisms in a bird flock or fish school.

The values chosen for algorithms' parameters are discussed in Section 5.4, which is dedicated to show the results and compare these algorithms.

5.3.1 Firefly Algorithm

The Firefly Algorithm (FA) is a metaheuristic optimization algorithm developed by Xin-She Yang [3] that mimics the behaviour of fireflies.

The main concept of such algorithm is to create a population of fireflies representing candidate solutions in a d -dimensional space for a certain optimization problem, where d is the number of variables to be optimized.

The fireflies are then attracted to each other following a set of rules:

- **Unisexuality**
All fireflies are unisex, so that one firefly is attracted to other fireflies regardless of their sex.
- **Attractiveness**
The attractiveness of a firefly is proportional to its brightness, thus for any two fireflies the less bright will move towards the brighter one. Both brightness and attractiveness decrease as the distance between two fireflies increases.
- **Brightness**
The brightness of a firefly is determined and affected by the objective function.

5.3.1.1 Algorithm Overview

Algorithm 2 Firefly Algorithm Approach

procedure FIREFLY ALGORITHM

Input: N, G, d, L, U

Output: A d -dimensional vector, i.e. the candidate solution (firefly) with best fit value

Initialization: generate N d -dimensional vectors $n_i, i \in [1, N]$

Light intensity I_i for n_i is determined by $f(n_i)$

Define light absorption coefficient σ

current_gen $\leftarrow 1$

while current_gen $< G$ **do**

for $i = 1, \dots, N$ **do**

for $j = 1, \dots, N$ **do**

if $I_i < I_j$ **then**

 Move firefly i towards j

 Vary attractiveness with distance r

 Evaluate new solutions and update light intensity

end for j

end for i

 Rank the fireflies and find the current global best g^*

end while

Parameter	Description
N	The population's size, i.e. the number of fireflies (candidate solutions) that move in the solution space to find a good solution.
G	The maximum generation. After reaching that stage of evolution, the algorithm stops.
d	The chromosome's dimension, i.e. the number of variables to be optimized by the algorithm.
L	A d-dimensional vector containing the lower bounds for each variable.
U	A d-dimensional vector containing the upper bounds for each variable.
Λ_0	Attractiveness at zero distance, described in Section 5.3.2.1
σ	Light absorption coefficient that influences the attractiveness distance, described in Section 5.3.2.2.
ρ	Randomization parameter influencing the fireflies' motion, described in Section 5.3.2.3

Table 4: Parameters for the genetic algorithm

5.3.1.2 Algorithm Description

The following sections serve the purpose of describing in depth the steps and the specifics of the Firefly Algorithm.

Section 5.3.1.3 describes how light intensity influences the relative attractiveness between two fireflies.

Section 5.3.1.4 describes how fireflies move in the solution space through the iterations.

Table 4 summarizes and describes the parameters used by the algorithm.

5.3.1.3 Light Intensity and Attractiveness

The brightness I of a firefly at a particular location depends on the fitness value returned by the fitness function at that specific location.

Each firefly follows a path that brings it closer to other attractive fireflies. Relative attractiveness Λ_{ij} however is relative, and it will vary with the distance r_{ij} between firefly i and j .

Equation 32 shows the relation between distance and attractiveness, where Λ_0 is the attractiveness at zero distance and σ is a parameter called *light absorption coefficient*.

$$\Lambda_{ij} = \begin{cases} \Lambda_0 e^{-\sigma r_{ij}^2} & \text{if } I_j > I_i \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

Equation 32 suggests that the brighter is a firefly, the more attractive it is to any other firefly. The light intensity I_j for the j -th firefly is determined by the fit function.

$$I_j = f_{\text{fit}}(n_j) = f_{\text{fit}}(n_{j,1}, n_{j,2}, \dots, n_{j,d}) \quad (33)$$

The fit function chosen for this work is the same as for the Genetic Algorithm approach, and is shown in Equation 29. The distance r_{ij} between firefly i and j at step k is given by the Euclidean distance between the two.

$$\begin{cases} r_{ij}^k &= \|n_j^k - n_i^k\| \\ &= \sqrt{(n_{j,1}^k - n_{i,1}^k)^2 + (n_{j,2}^k - n_{i,2}^k)^2 + \dots + (n_{j,d}^k - n_{i,d}^k)^2} \end{cases} \quad (34)$$

5.3.1.4 Firefly Movement

The movement of a firefly n_i towards a more attractive n_j at step $k \in [1, G]$ is given by Equation 35.

$$n_i^k = n_i^k + \Lambda_{ij}^k (n_j^k - n_i^k) + \rho v_i^k \quad (35)$$

Here Λ_{ij} is the attractiveness of firefly j to firefly i , ρ is a *randomization parameter* and v_i is a vector of d random numbers drawn from a Gaussian or uniform distribution.

It is important to notice that the movement of a given firefly towards a less attractive one is given only by a random factor, but it is still present. This means that the most attractive fireflies still move and can potentially find better solutions.

After the fireflies' positions are updated, new fitness values are calculated. New generations are obtained by using the same set of equations. Over successive generations, the fireflies converge to local and global optimal values, completing the optimization process.

5.3.2 Modified Firefly Algorithm

A modified version of the Firefly Algorithm has been developed and studied by M.A. Zaman and U. Sikder [19]. The main idea behind the Modified Firefly Algorithm is to change the *process control parameters* Λ_0 , σ and ρ through the iterations. Typical values of these parameters are $\Lambda_0 = 1$, $\sigma = 1$ and $\rho \in [0.1, 0.2]$. By changing these from static to dynamic parameters it is possible to perform a more precise tuning of the fireflies behaviour, reflecting thus on the performance of the algorithm on finding their optimal values.

5.3.2.1 Attractiveness at Zero Distance

It has been studied that the attractiveness at zero distance parameter Λ_0 controls the *exploration distance* of the fireflies: higher values of

Λ_0 encourage global exploration while lower values encourage close refinement (local exploitation).

Hence it would be better to make Λ_0 a decreasing value over the iterations, such that initially the fireflies explore a wide range of possible solutions minimizing the risk of getting stuck in a local optimum, while near the end of the execution a lower value would allow for fine refinement around the different optima found.

The formula for the attractiveness parameter for iteration $k \in [1, G]$ has been changed as shown in Equation 36.

$$\Lambda_0(k) = a - b \frac{1 - k}{1 - G} \quad (36)$$

From Equation 36 it is possible to notice that the value of Λ_0 goes from a (when $k = 1$) to $a - b$ (when $k = G$).

5.3.2.2 Light Absorption Coefficient

The value of σ control the effect of distance on the motion of fireflies. Equation 32 suggests that lower values of σ will cause movement towards other attractive fireflies even if they are far away. This may lead to a premature convergence of a high number of fireflies towards a local maximum.

However, after a sufficient number of generations, it is desirable that the fireflies start to converge towards the maximum value found so far. To do so, a lower value of σ is required.

The formula for the *light absorption coefficient* σ for iteration k has been changed as shown in Equation 37.

$$\sigma(k) = 2 \cdot \left(7^{-k/G} \right), \quad c \geq 0, d \geq 0 \quad (37)$$

5.3.2.3 Randomization Parameter

The *randomization parameter* ρ controls the random component of the firefly's motion. A large value of ρ encourages broad random exploration. Using the same arguments as for the *light absorption coefficient*, it can be inferred that a gradual decrease of the value of ρ over successive iterations should improve the performance of the algorithm as the fireflies must explore the solution space widely at the beginning to find good candidate solutions and then refine them in latter steps.

The formula for the *randomization parameter* ρ for iteration k has been changed as shown in Equation 38.

$$\rho(k) = 0.1 \cdot e^{-4.8k/G} \quad (38)$$

The proposed modification of the algorithm using dynamic parameters are expected to provide a good mix of global exploration and local exploitation, resulting in high accuracy and fast convergence of the algorithm towards the global optimum.

5.3.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [4] [20].

PSO is similar to a genetic algorithm, meaning that both algorithms are initialized with a population of random solutions. Unlike a GA, however, candidate solutions, called *particles*, are also assigned a randomized velocity. After that, they are flown into the problem space.

Each point of the problem space represents a set of parameters for the optimization problem that has to be solved, in this case the Bouc-Wen model parameter search. Each particle keeps track of its coordinates in the problem space, and remembers the coordinate of the point which has yielded the best solution, in other words the point that has the highest fitness value found by them so far. This value is called *pbest*, which means *personal best*.

Another value is tracked and known by all particles, which is the *global best* value found by any particle up until that certain iteration of the algorithm. This value is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity (accelerating) each particle towards its *pbest* and *gbest* locations. Acceleration is weighted by a random term, with separate random numbers generated to dictate acceleration towards *pbest* and *gbest* locations.

5.3.3.1 Algorithm Overview

Algorithm 3 Particle swarm optimization approach

procedure PARTICLE SWARM OPTIMIZATION

Input: N, G, d, L, U, V_{\max}

Output: A d -dimensional vector, i.e. the candidate solution (particle) with best fit value

Initialization: generate N d -dimensional vectors $n_i, i \in [1, N]$

current_gen $\leftarrow 1$

while current_gen $< G$ **do**

Evaluation: compute the fit value $\text{fit}(i)$ for each vector

Pbest comparison: Compare the particle's fit value with particle's *pbest*. If the current value is better than *pbest*, then set *pbest* value equal to the current value, and *pbest* location equal to the particle's current location in the d -dimensional space

Gbest comparison: Compare the particle's fit value with *gbest*. If the current value is better than *gbest*, then set *gbest* value equal to the current value, and *gbest* location equal to the particle's current location in the d -dimensional space

Update: change the particle's velocity and position

end while

5.3.3.2 Algorithm Description

The following sections serve the purpose of describing in depth the steps and the specifics of the Particle Swarm Optimization algorithm.

Section 5.3.3.3 describes how the values for velocity and position for the particles are updated each iteration.

Section 5.3.3.4 describes the purpose of and how to compute the inertia weight w used when calculating the speed of a particle.

Section 5.3.3.5 describes the purpose of the acceleration constants c_1 and c_2 .

5.3.3.3 Velocity and Position Update

Given x_i and v_i which are respectively the i -th particle's position and velocity, then the movement of the i -th particle is dictated by Equations 39 and 40. In Equation 39, w is the *inertia weight*, x_{pb} is the position associated to the particle's personal best value, and x_{gb} is the position associated to the global best value among all particles.

$$v_i = w \cdot v_i + c_1 \cdot \text{rand}() (x_{pb} - x_i) + c_2 \cdot \text{rand}() (x_{gb} - x_i) \quad (39)$$

$$x_i = x_i + v_i \quad (40)$$

Particles' velocity are constrained by the maximum velocity parameter V_{\max} . This is a very important parameter, as it determines the fineness with which regions between the present position and the best positions known are searched. If V_{\max} is too high, particles may overshoot good solutions. On the other hand, if V_{\max} is too low, particles may move very slowly and may not be able to find a good solution before the algorithm comes to a halt.

5.3.3.4 Inertia Weight

The use of a *inertia weight* w has provided improved performance in a number of applications, as an opportune selection of the inertia weight provides a balance between global and local exploration, and results in fewer iterations on average to find a sufficiently good solution.

The *inertia weight* w changes dynamically with the current iteration of the algorithm according to the following law:

$$w(k) = a - (a - b) \cdot \frac{k}{G}, \quad k \in [1, G]$$

5.3.3.5 Acceleration Constants

The *acceleration constants* c_1 and c_2 represent the importance of the personal and global best according to the particles. If c_1 is greater than c_2 , the particles will tend to judge as better their own personal best and will be inclined to move towards that location.

5.4 ALGORITHM COMPARISON

Simulations have been run on all the algorithms proposed in the previous sections. Each one of them has been tested with both the standard and the generalized Bouc-Wen model of hysteresis.

Parameter	Description
N	30
G	30
d	4
L	[0.5, 0.5, -0.5, -0.1]
U	[1, 0.99, 0.5, 0.1]
P_{cross}	0.5
P_{mut}	0.1
m_m	0.05

Table 5: Parameters for the genetic algorithm

Since the generalized version of the hysteresis component consists of 8 variables it is safe to assume beforehand that the execution time of the algorithms would be higher. A comprehensive comparison of the execution times for the algorithms is described in Section 5.4.3.

5.4.1 System Specifications

5.4.2 Solution Comparison

This Section serves the purpose of comparing the solutions found using the algorithm previously described when trying to find suitable parameters both for the classic and the generalized Bouc-Wen model of hysteresis.

5.4.2.1 Genetic Algorithm – Classic Bouc-Wen Model

Table 5 contains the parameters used for the genetic algorithm approach. Those parameters have been used for the genetic algorithm using all three kinds of crossover strategies: random, weighted and proportional.

Figures 9, 10 and 11 show the trend of the classic Bouc-Wen model parameters A , α , β and γ depending on the crossover strategy used.

It is important to note that due to how the three strategies have been developed, the performance of the random and weighted methods depend strictly on the initial population, as the individuals' chromosomes are always only swapped. The final result will approach the best possible solution attainable using the chromosomes of the initial population. However, it is certainly possible that the final fit may not be acceptable.

Due to this fact, the proportional crossover strategy proved to be the best as it generates new chromosomes (and thus, completely new individuals) each generation.

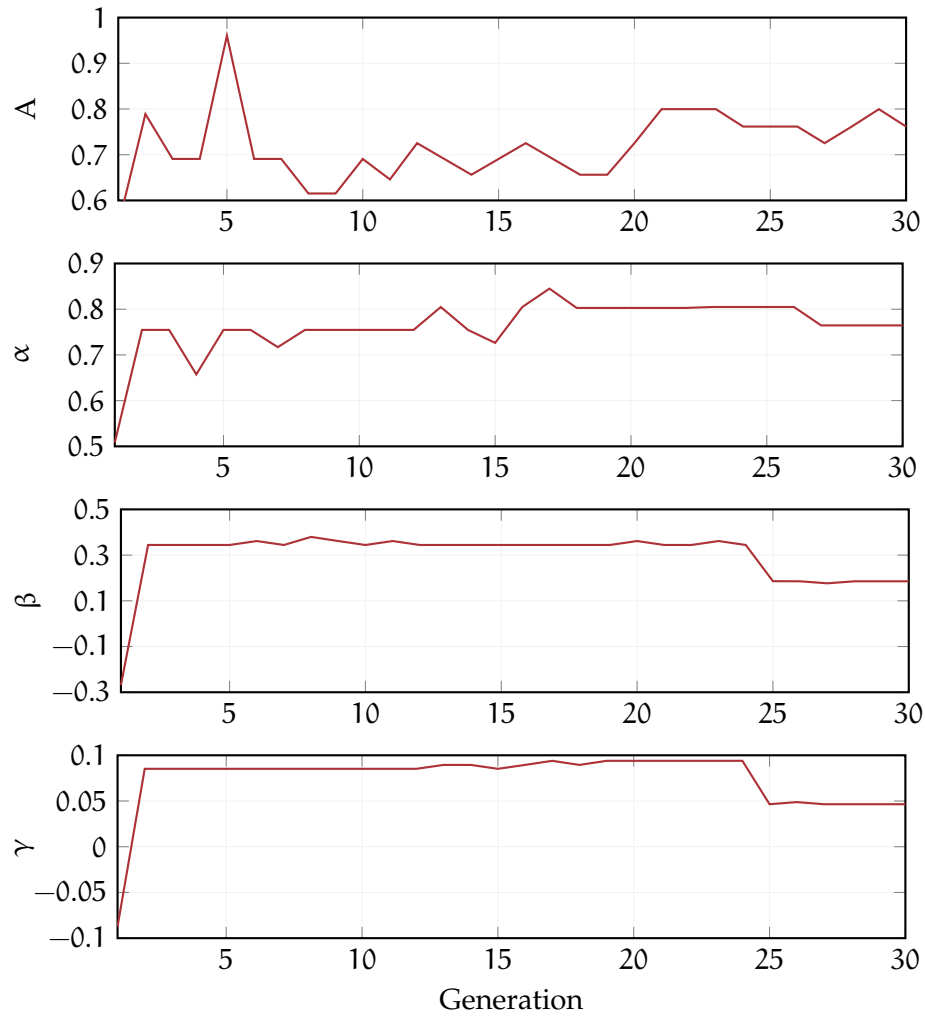


Figure 9: Genetic Algorithm – Classic BW Model (Random crossover)

A series of sharp changes is noticeable in Figure 9. As said beforehand, this behaviour is due to the fact that the set of chromosomes obtainable by the individual in the population is fixed to the initial one.

Nonetheless it is possible to notice that the variables β and γ are the first to settle to fairly consistent values, up until generation 25. This behaviour is probably due to the current best individual being near a local optimum.

The final values for each parameter using this crossover strategy are in Table 6.

Parameter	A	α	β	γ
Final value	0.76158	0.76455	0.18539	0.04647

Table 6: Random crossover – Final values (Classic BW model)

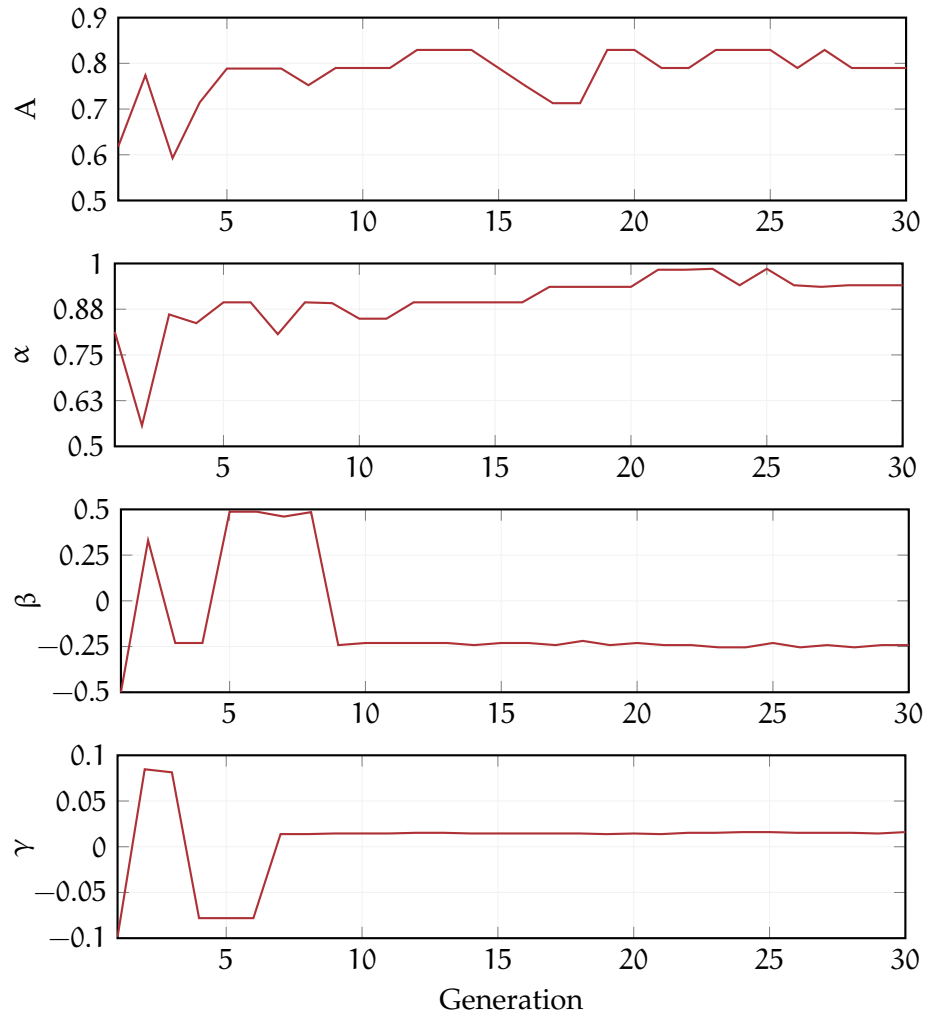


Figure 10: Genetic Algorithm – Classic BW Model (Weighted crossover)

As for the random crossover strategy, the weighted crossover strategy operates on the closed set of initial chromosome values, but combines them judging the fit value of the individuals instead of doing it randomly.

The most important difference between these two results is the fact that it takes fewer generations to reach what may be a global optimum. This is especially noticeable in Figure 10, as the parameters β and γ stabilize on similar values around generation 10 and 7, respectively.

The final values for each parameter using this crossover strategy are in Table 7.

Parameter	A	α	β	γ
Final value	0.78982	0.94050	-0.24214	0.01601

Table 7: Weighted crossover – Final values (Classic BW model)

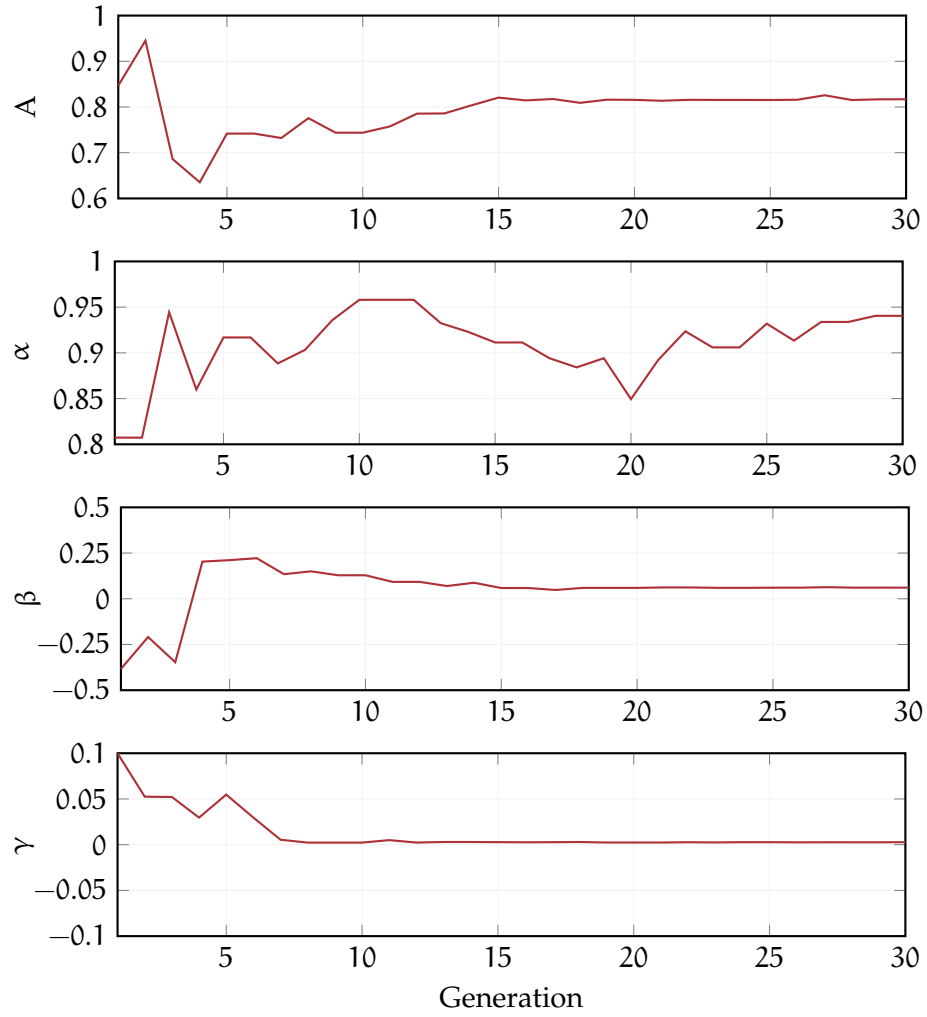


Figure 11: Genetic Algorithm – Classic BW Model (Proportional crossover)

In contrast to the parameter trends of random and weighted crossover strategies, the proportional crossover strategy shows a smoother trend. This is especially noticeable for parameters A , β and γ .

As for the weighted crossover method, it is possible to see an almost immediate setting of parameters β and γ , as it happens for both around generation 7.

The final values for each parameter using this crossover strategy are in Table 8.

Parameter	A	α	β	γ
Final value	0.81699	0.94050	0.06078	0.00273

Table 8: Weighted crossover – Final values (Classic BW model)

Figure 12 shows how the hysteresis loop shows up with the final parameters present in Table 8.

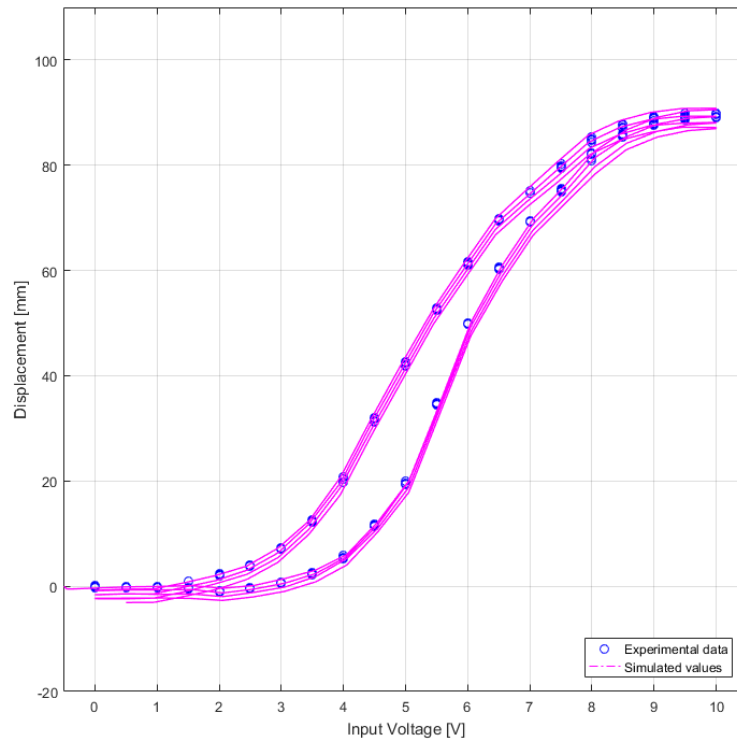


Figure 12: Hysteresis loop result using genetic algorithm's parameters (proportional crossover approach)

5.4.2.2 Modified Firefly Algorithm – Classic Bouc-Wen Model

Table 9 contains the parameters used for the modified firefly algorithm approach.

Parameter	Description
N	30
G	30
d	4
L	[0, 0.5, -0.5, -0.5]
U	[2, 0.99, 0.5, 0.5]
α	1.1
b	0.3

Table 9: Parameters for the genetic algorithm

Figure 13 shows the trend of Bouc-Wen parameters A , α , β and γ using the Modified Firefly Algorithm.

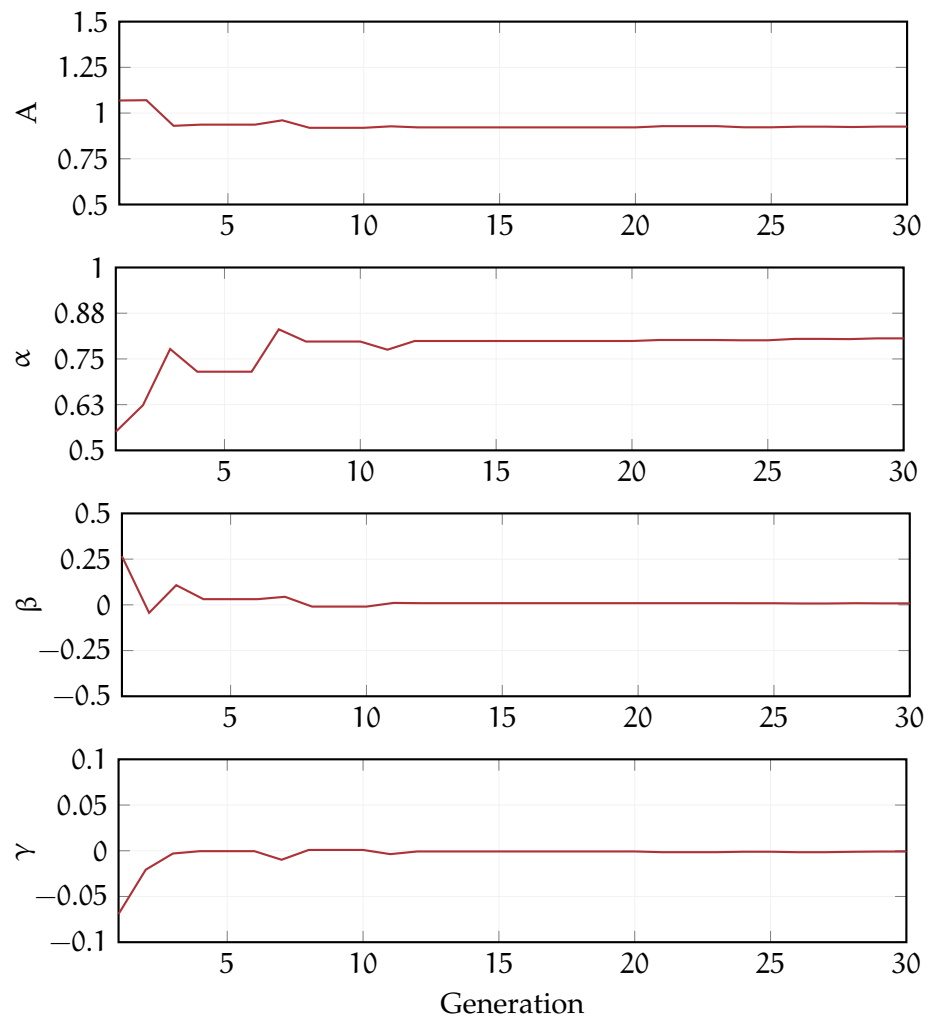


Figure 13: Modified Firefly Algorithm – Classic BW Model

5.4.3 Time Comparison

CONCLUSIONS AND FUTURE WORK

Part II

APPENDICES

SCRIPTS AND ALGORITHMS

This Appendix contains all scripts and algorithms used for this work.

A.1 SCRIPTS

A.1.1 *Stair Input Generation*

This script generates the step wave that is to be given as input to the input and output proportional valves. Given that they have to be symmetric, the wave given to the output valve is defined as follows.

$$y' = 10 - y$$

Stair input generation

```

1 close all
  clc
  step = 1;      % Volts between two consecutive steps
  y = zeros(1,50*10*(1/step));
  for v = 0:step:10
6      y((v*50)*(1/step)+1:(v+step)*(1/step)*50) = v;
  end
  y = [y fliplr(y)];
  y = [y y y y]';
  last = (length(y)-1)/10;
11 t = (0:0.1:last)';
  y = [t y];

```

A.2 OPTIMIZATION ALGORITHMS

A.2.1 *Genetic Algorithm*

A.2.2 *Modified Firefly Algorithm*

A.2.3 *Particle Swarm Optimization*

BOUC-WEN MODEL OF HYSTERESIS

BIBLIOGRAPHY

- [1] B. Tondu and P. Lopez, "Modeling and control of McKibben artificial muscle robot actuators," *IEEE control systems*, vol. 20, no. 2, pp. 15–38, 2000.
- [2] P. J. Fleming and R. Pursehouse, "Genetic algorithms in control systems engineering," 2001.
- [3] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
- [4] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, Springer, 2011, pp. 760–766.
- [5] C. P. Chou and B. Hannaford, "Measurement and modeling of McKibben pneumatic artificial muscles," *IEEE Transactions on robotics and automation*, vol. 12, no. 1, pp. 90–102, 1996.
- [6] F. Al-Bender, V. Lampaert, and J. Swevers, "The generalized Maxwell-slip model: a novel model for friction simulation and compensation," *IEEE Transactions on automatic control*, vol. 50, no. 11, 2005.
- [7] D. Lederer, H. Igarashi, A. Kost, and T. Honma, "On the parameter identification and application of the Jiles-Atherton hysteresis model for numerical modelling of measured characteristics," *IEEE Transactions on magnetics*, vol. 35, no. 3, pp. 1211–1214, 1999.
- [8] P. Ge and M. Jouaneh, "Generalized Preisach model for hysteresis nonlinearity of piezoceramic actuators," *Precision engineering*, vol. 20, no. 2, pp. 99–111, 1997.
- [9] R. Bouc, "Modèle mathématique d'hystérésis : application aux systèmes à un degré de liberté," *Acustica*, 1969.
- [10] Y. K. Wen, "Method for random vibration of hysteretic systems," *Journal of the engineering mechanics division*, vol. 102, no. 2, pp. 249–263, 1976.
- [11] C. P. Chou and B. Hannaford, "Static and dynamic characteristics of mckibben pneumatic artificial muscles," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, 1994.
- [12] H. F. Schulte, "The Characteristics of the McKibben Artificial Muscle," in *The Application of External Power in Prosthetic and Orthotics*, National Academy of Sciences - National Research Council, Ed., 1961.
- [13] *System identification toolbox, Create linear and nonlinear dynamic system models from measured input-output data*, 2018. [Online]. Available: <https://uk.mathworks.com/products/sysid.html>.

- [14] F. Preisach, "Über die magnetische nachwirkung," *Zeitschrift für Physik*, vol. 94, no. 5, pp. 277–302, 1935, ISSN: 0044-3328. DOI: [10.1007/BF01349418](https://doi.org/10.1007/BF01349418). [Online]. Available: <https://doi.org/10.1007/BF01349418>.
- [15] F. Al-Bender, V. Lampaert, and J. Swevers, "The generalized maxwell-slip model: A novel model for friction simulation and compensation," *IEEE Transactions on automatic control*, vol. 50, no. 11, pp. 1883–1887, 2005.
- [16] M. A. Krasnosel'skii and A. V. Pokrovskii, *Systems with hysteresis*. Springer Science & Business Media, 2012.
- [17] M. Ismail, F. Ikhoulane, and J. Rodellar, "The hysteresis bouc-wen model, a survey," *Archives of Computational Methods in Engineering*, vol. 16, no. 2, pp. 161–188, 2009.
- [18] J. Song and A. Der Kiureghian, "Generalized bouc-wen model for highly asymmetric hysteresis," *Journal of engineering mechanics*, vol. 132, no. 6, pp. 610–618, 2006.
- [19] M. A. Zaman and U. Sikder, "Bouc-wen hysteresis model identification using modified firefly algorithm," *Journal of Magnetism and Magnetic Materials*, vol. 395, pp. 229–233, 2015.
- [20] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, IEEE, 1995, pp. 39–43.